When Measurement Benefits the Measured

by Mark Kasunic and William Nichols

04.23.14 • 1:30 pm ET–2:30 pm ET

Mark Kasunic
Senior Member of the Technical Staff
Software Engineering Institute

Mark Kasunic is a senior member of the technical staff at the Software Engineering Institute (SEI) at Carnegie Mellon University. He is currently a member of the Team Software Process Initiative within the Software Solutions Division. Since joining the SEI in 1994, his work has focused on transitioning performance improvement technologies into practice through applied research, course development, coaching, and training. His current research and development interests include data quality assessment and improvement, project performance measurement, and practical measurement and analysis approaches that help individuals and teams improve their technical performance. Mark has an extensive list of technical publications and conference presentations addressing software engineering and measurement. Before joining the SEI, Mark was an engineer and manager at Boeing in Seattle. He has a Masters Degree in Systems Engineering and is a senior member of IEEE. Mark is a certified TSP Mentor Coach and a certified Scrum Master.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
23 APR 2014

2. REPORT TYPE

3. DATES COVERED
00-00-2014 to 00-00-2014

4. TITLE AND SUBTITLE
When Measurement Benefits the Measured

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

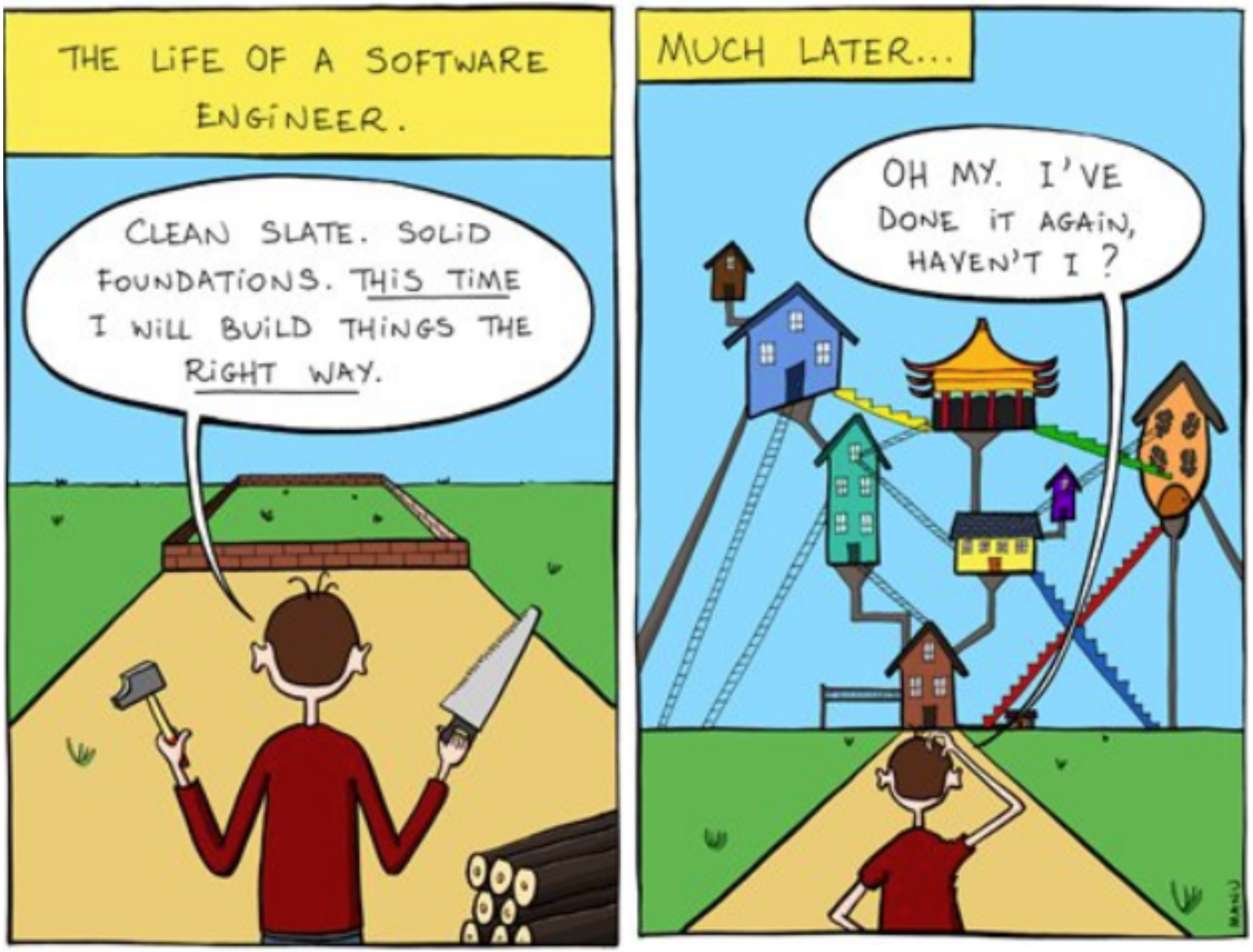
12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**105**	


THE LIFE OF A SOFTWARE ENGINEER.
CLEAN SLATE. SOLID FOUNDATIONS. THIS TIME I WILL BUILD THINGS THE RIGHT WAY.
MUCH LATER...
OH MY. I'VE DONE IT AGAIN, HAVEN'T I ?
MANU

The Sobering State of Software Engineering

39% of software projects are successful

43% of software projects cost more, take longer, or do less

18% of software projects failed

The 2013 Chaos Manifesto – The Standish Group - http://versionone.com/assets/img/files/CHAOSManifesto2013.pdf

Another Data Point

In a survey of 166 IT leaders:

89% of projects do not regularly meet their budget

59% projects are typically delivered late

33% state that rework is at least 25% of their budget

2014 IT Leadership Survey - Blueprint Software Systems Inc.

PROBLEM?

Can Measurement Help?

> CEO’s have a lower opinion of software groups than of other technical groups due to consistently optimistic estimates, schedule delays, cost overruns, poor quality when delivered, and outright failures. Software is much worse in all of these.
>
> Better measures of projects … will improve the professional status of the software community and perhaps lead to CEO’s having more respect for software groups than they have today.

Capers Jones
InfoQ Interview
March 30, 2014
http://www.infoq.com/articles/Jones-measuring-agile-adoption

The World Without Measurement

Science?

Engineering?

Medicine?

We All Measure ….

Driving?

Clothing Size?

Cooking?

AM
PM
LA:TE
ALARM
FM 88 92 96 100 104 108 MHz FM/AM CLOCK RADIO
AM 53 60 70 80 100 130 170 x10kHz

Getting to Work on Time?

How We Use Measurement

Gerald (Jerry) Weinberg

First-Order Measurement

What *seems* to be happening?

Tends to be qualitative and fast.

Second-Order Measurement

What's *really* happening? And how is it changing?

It needs to be quantitative; subject to more refined models.

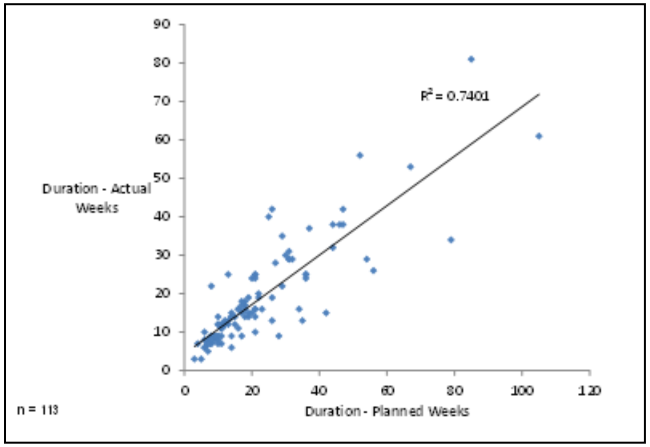
Third-Order Measurement

What *happens* in a more general and universal sense?

Needs to be precise with checks for validity; statistical variation must be characterized and interpreted appropriately.

First-Order Measurement

Second-Order Measurement



Third-Order Measurement

The State of the Practice

An Issue

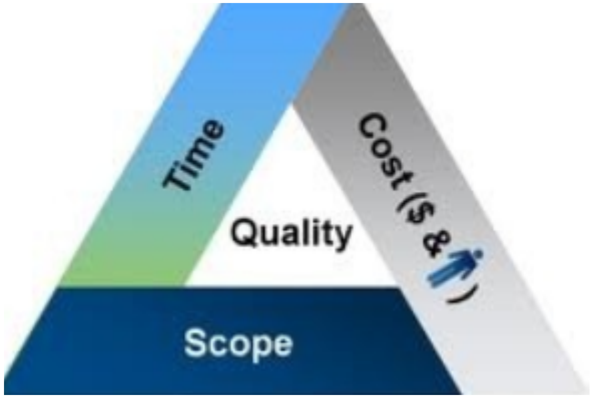
The results of applying many software development methods are unpredictable.

Decision making about method selection is based on suppositions, opinions, and fads.

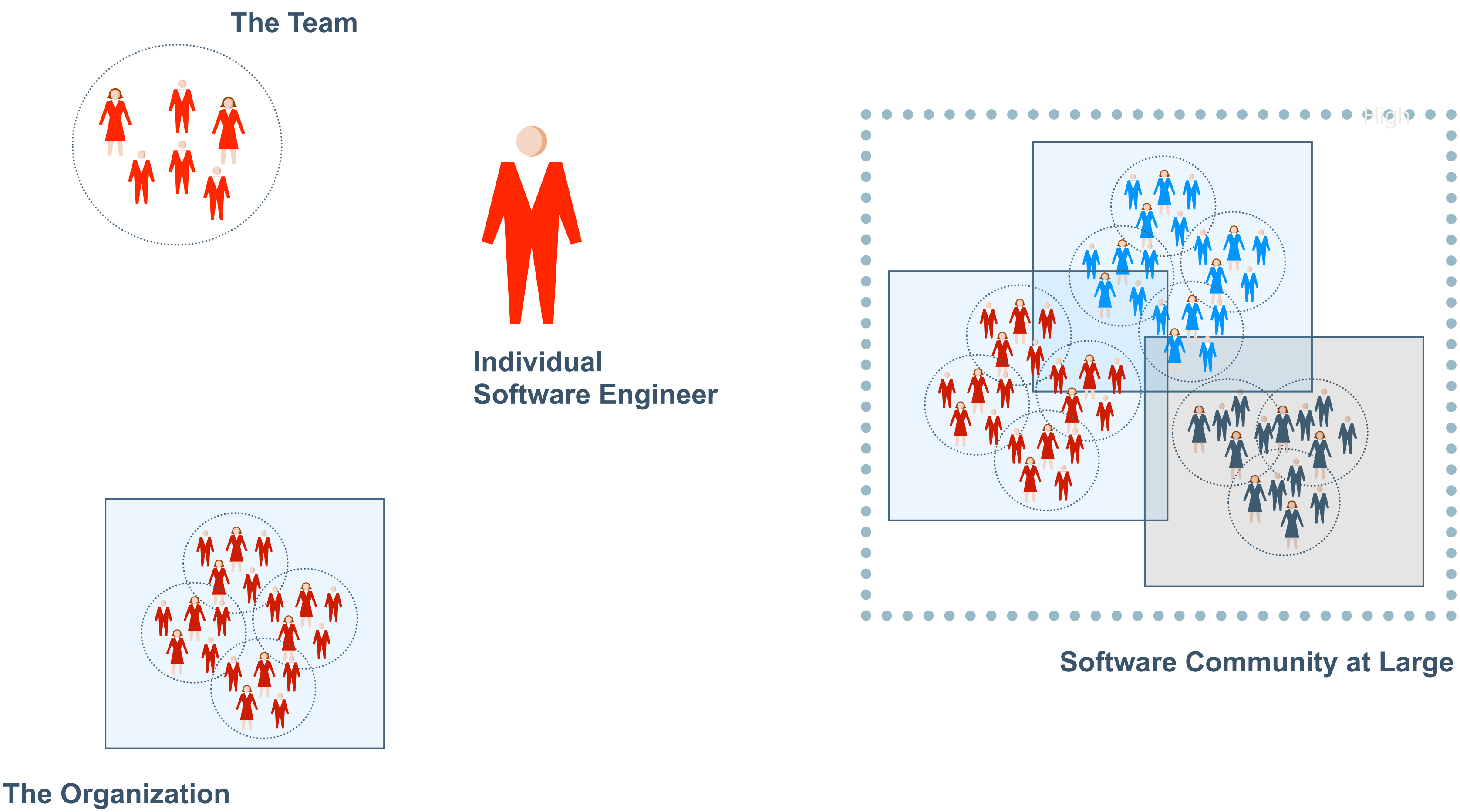
What We Need

We need to set aside perceptions and market-speak … and transform software engineering into an engineering discipline.

Decisions should be based on fair and unbiased analysis of information.



Measurement in Your Work Life



The Team

Individual
Software Engineer

Software Community at Large

The Organization

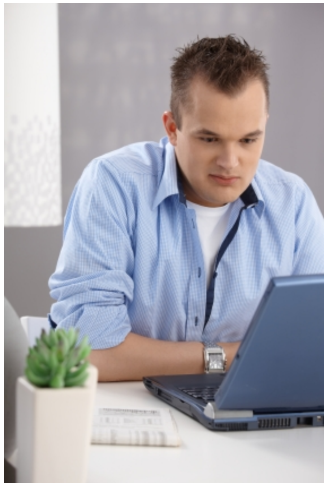
Measurement & the Individual Software Engineer

You

Stellar athletes understand that they must set specific goals to reach their potential.

Measurement provides the necessary feedback that drives athletes to achieve world-class excellence.

Can software engineers leverage goal-setting and measurement in the same way?



Do you?

What type of measures do you *typically* use to estimate the duration of your work for a schedule?

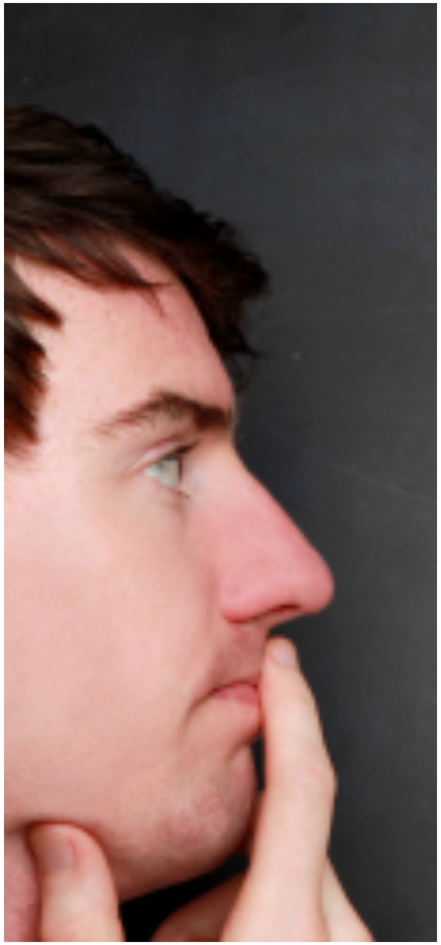
□ First order: Qualitative - based on what I think I’ve done before.

□ Second order: Quantitative - based on quantitative data from previous project(s).

□ Third order: Statistical - based on statistical patterns of data from my previous projects.

□ None of the above.

Should You Be Using Measurement?



Yes. And it needs to go beyond first order measurement.

Measurement is needed to **manage** your work.

Managing the Work

But isn't it the managers job to *manage*?

The Evolution of the Management Approach

Body Management

People as oxen.

Frederick Taylor

Task Management

People as machines.

Peter Drucker

Knowledge Management

People as individuals.

Evolution of the Worker

Hunter-Gatherer

Farmer & Artisan

Industrial Revolution Worker

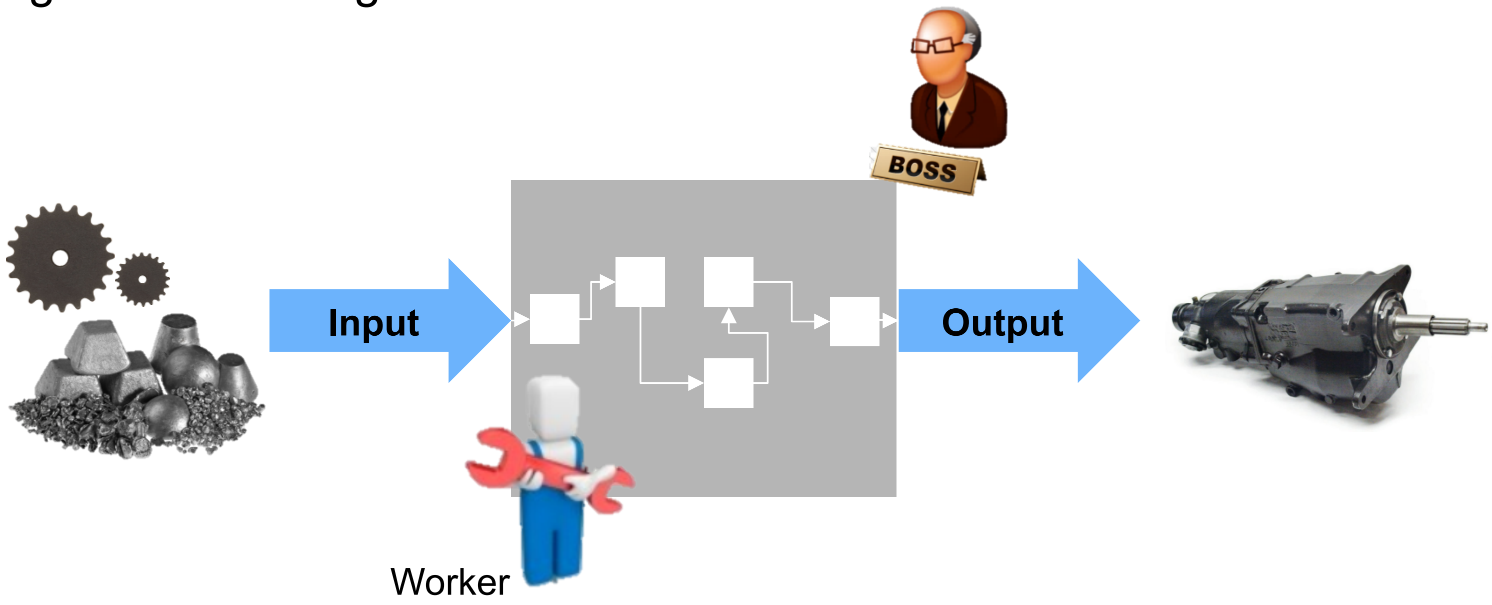
Technology Professional

Taylorism – Scientific Management

For years, the basic power equation in organizations was simple and effective:

Knowledge held by a few (the managers), plus iron discipline over the many (the workers).

The worker was viewed as an instrument, a bundle of muscles programmed through instruction.



The Birth of the Knowledge Worker

The Technology Professional

- New data processing age was born during 2nd half of 20th century
- Work became asynchronous and non-linear
- Nature of knowledge work demanded significant control by the worker (instead of the manager)

What Differentiates Knowledge Work?

Manual work

Consists of converting materials form one form to another.

The work output is tangible.

Knowledge work

The work is done in the head.

The work can’t be seen.

A Shift In the Locus of Control

This new breed of worker has a new job: converting knowledge into actions that convert information from one form to another.

Because the behaviors of a knowledge worker are primarily private, supervisors cannot supervise.

Due do the nature of knowledge work, it is the worker that has almost total authority in matching methods to the varying job tasks and situations that they encounter.

However, with this reality, there is also a shift in responsibility …

Managing the Work

But isn't it the managers job to *manage*?

No.

Management provides goals.

Knowledge workers manage their work.

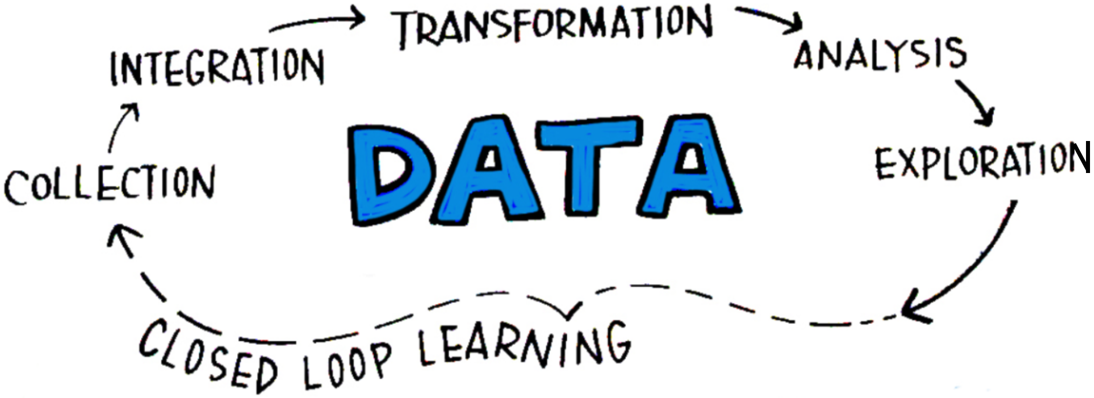
Controlling Your Own Destiny

To control the way they work, software engineers must plan their projects.

For management to trust these plans, the engineers must make accurate plans.

To make accurate plans, they must have data.

To have data, they must measure their performance.

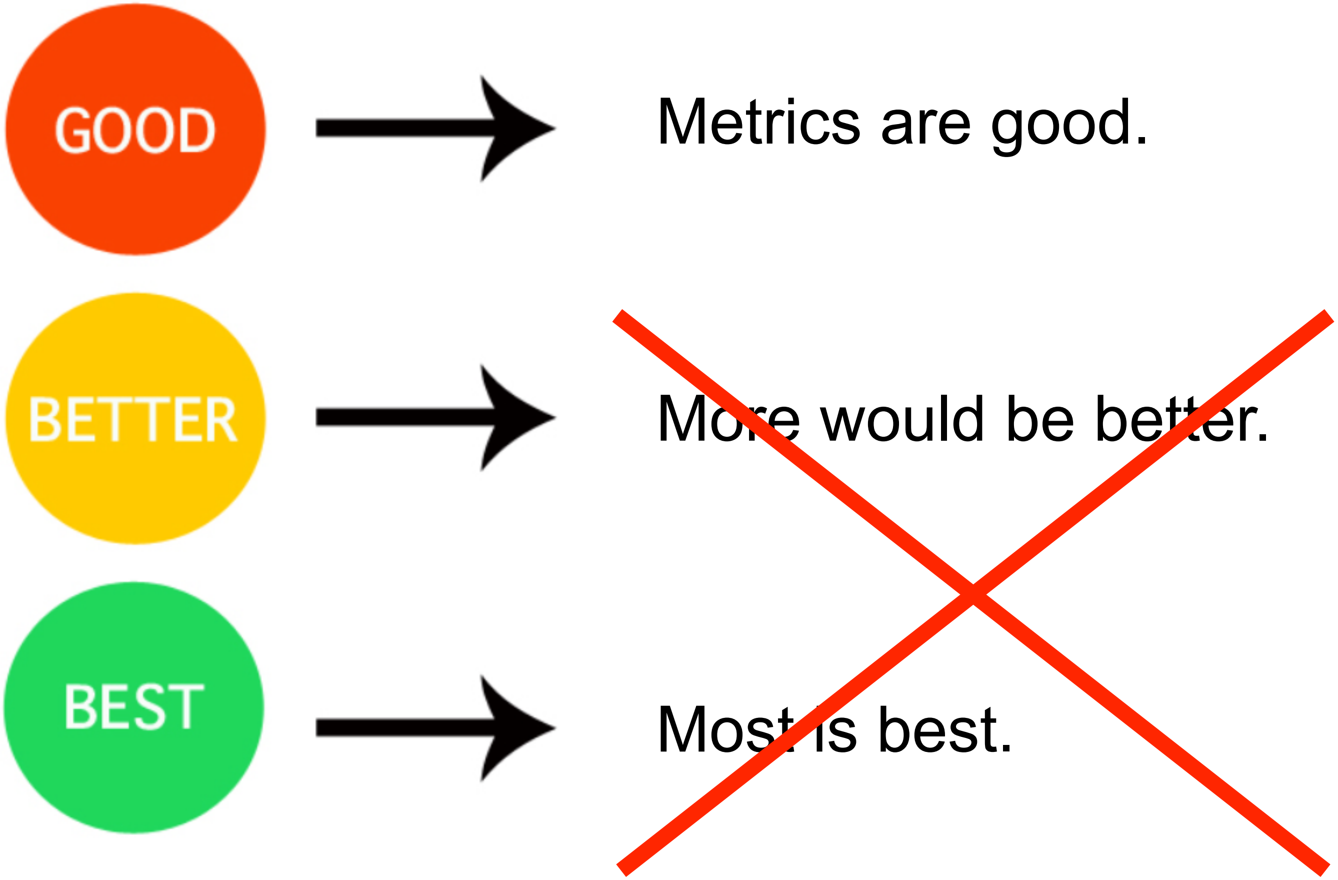


You Need Data To Manage Yourself

Metrics are good.

But …

This Is A Data Collection Fallacy



GOOD → Metrics are good.

BETTER → ~~More would be better.~~

BEST → ~~Most is best.~~

Only Four Basic Measures Needed

Software engineers only need to collect four basic measures to mange their schedule performance and the quality of their work.

Time on Task

Size

Defects

Schedule

Measures Are Estimated and Then Tracked

At the beginning of an effort, the work is planned and divided into a *set* of tasks or activities called *phases.* The basic measures are *estimated.*

- product size
- time-in-phase
- defects injected into a phase
- defects found in a phase
- task completion dates

During the project, these measures are collected in real time

- time-in-phase
- defect type injected in phase
- find/fix time for each defect
- task completion dates

When a product has been completed

- product size is measured

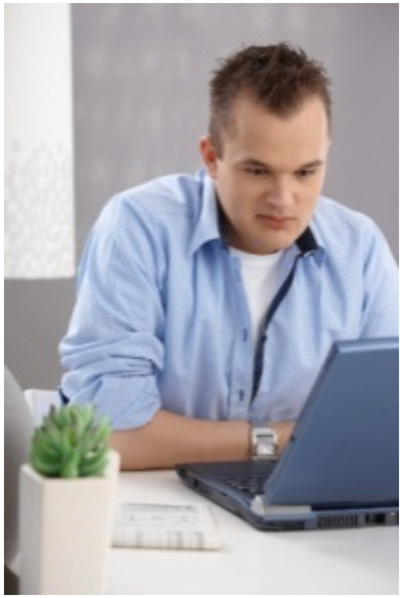
No one wants to be measured by others.

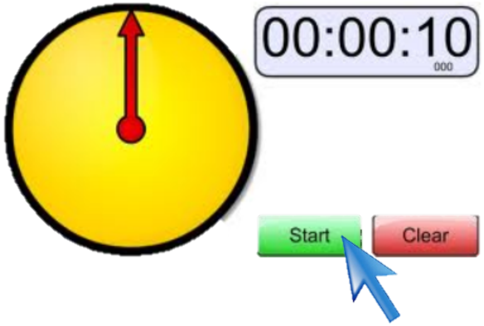
And, that’s not what we’re talking about here.

Tool Support For Data Collection

Collecting the four core measures would be impractical in the absence of software support.

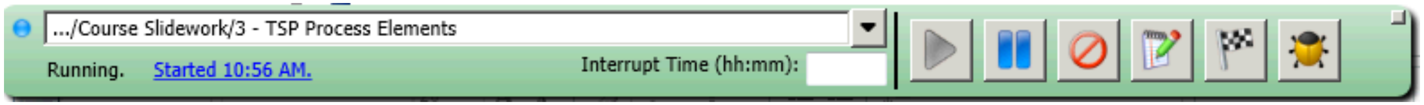
A number of tools are available that make it easy to collect this type of data.





Software Process Dashboard: http://www.processdash.com/

Collecting Personal Data

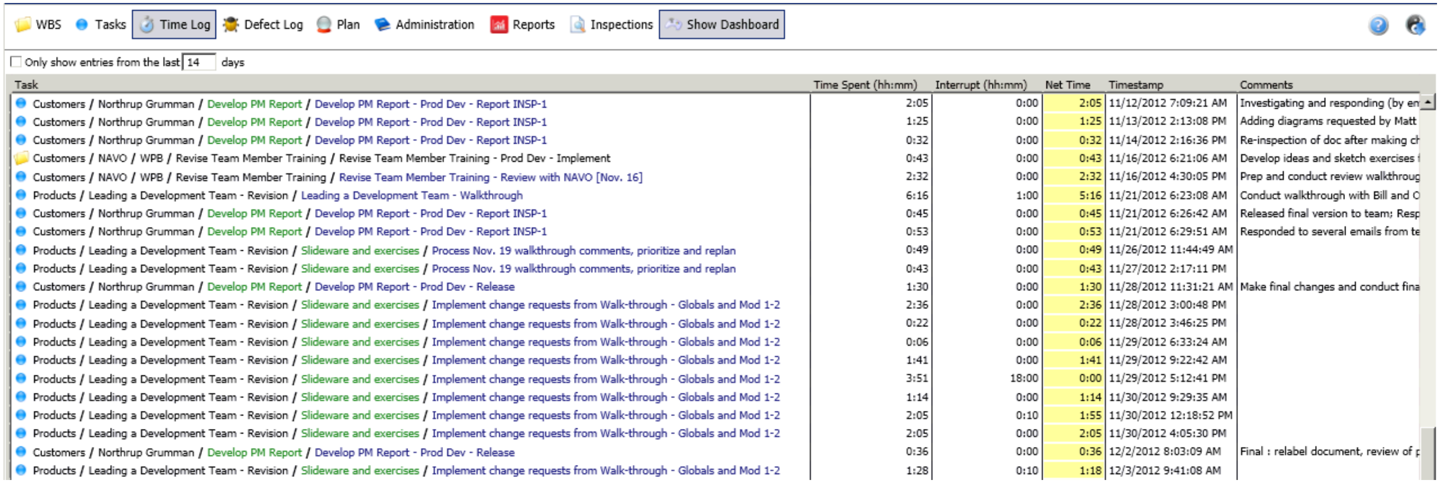

.../Course Slidework/3 - TSP Process Elements

Running. Started 10:56 AM.

Interrupt Time (hh:mm):

Automatically entered into *time log.*

Click to enter defect into log



□ Only show entries from the last 14 days

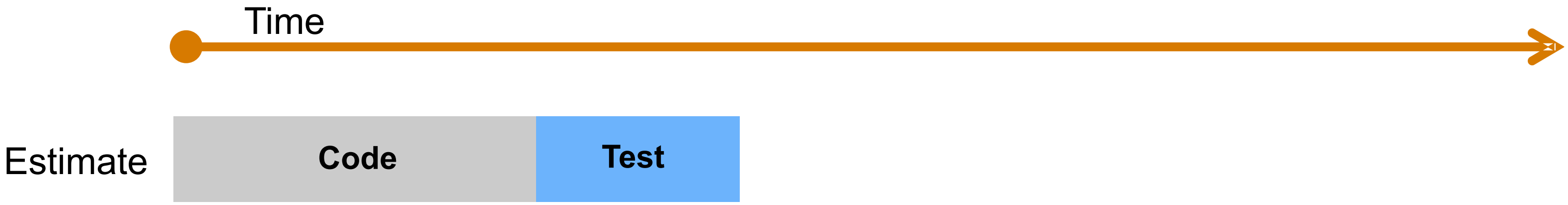
Task	Time Spent (hh:mm)	Interrupt (hh:mm)	Net Time	Timestamp	Comments
Customers / Northrup Grumman / Develop PM Report / Develop PM Report - Prod Dev - Report INSP-1	2:05	0:00	2:05	11/12/2012 7:09:21 AM	Investigating and responding (by em
Customers / Northrup Grumman / Develop PM Report / Develop PM Report - Prod Dev - Report INSP-1	1:25	0:00	1:25	11/13/2012 2:13:08 PM	Adding diagrams requested by Matt
Customers / Northrup Grumman / Develop PM Report / Develop PM Report - Prod Dev - Report INSP-1	0:32	0:00	0:32	11/14/2012 2:16:36 PM	Re-inspection of doc after making ch
Customers / NAVO / WPB / Revise Team Member Training / Revise Team Member Training - Prod Dev - Implement	0:43	0:00	0:43	11/16/2012 6:21:06 AM	Develop ideas and sketch exercises f
Customers / NAVO / WPB / Revise Team Member Training / Revise Team Member Training - Review with NAVO [Nov. 16]	2:32	0:00	2:32	11/16/2012 4:30:05 PM	Prep and conduct review walkthroug
Products / Leading a Development Team - Revision / Leading a Development Team - Walkthrough	6:16	1:00	5:16	11/21/2012 6:23:08 AM	Conduct walkthrough with Bill and O
Customers / Northrup Grumman / Develop PM Report / Develop PM Report - Prod Dev - Report INSP-1	0:45	0:00	0:45	11/21/2012 6:26:42 AM	Released final version to team; Resp
Customers / Northrup Grumman / Develop PM Report / Develop PM Report - Prod Dev - Report INSP-1	0:53	0:00	0:53	11/21/2012 6:29:51 AM	Responded to several emails from te
Products / Leading a Development Team - Revision / Slideware and exercises / Process Nov. 19 walkthrough comments, prioritize and replan	0:49	0:00	0:49	11/26/2012 11:44:49 AM	
Products / Leading a Development Team - Revision / Slideware and exercises / Process Nov. 19 walkthrough comments, prioritize and replan	0:43	0:00	0:43	11/27/2012 2:17:11 PM	
Customers / Northrup Grumman / Develop PM Report / Develop PM Report - Prod Dev - Release	1:30	0:00	1:30	11/28/2012 11:31:21 AM	Make final changes and conduct fina
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	2:36	0:00	2:36	11/28/2012 3:00:48 PM	
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	0:22	0:00	0:22	11/28/2012 3:46:25 PM	
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	0:06	0:00	0:06	11/29/2012 6:33:24 AM	
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	1:41	0:00	1:41	11/29/2012 9:22:42 AM	
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	3:51	18:00	0:00	11/29/2012 5:12:41 PM	
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	1:14	0:00	1:14	11/30/2012 9:29:35 AM	
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	2:05	0:10	1:55	11/30/2012 12:18:52 PM	
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	2:05	0:00	2:05	11/30/2012 4:05:30 PM	
Customers / Northrup Grumman / Develop PM Report / Develop PM Report - Prod Dev - Release	0:36	0:00	0:36	12/2/2012 8:03:09 AM	Final : relabel document, review of p
Products / Leading a Development Team - Revision / Slideware and exercises / Implement change requests from Walk-through - Globals and Mod 1-2	1:28	0:10	1:18	12/3/2012 9:41:08 AM	

Tracking Your Time

All activities that contribute to the value chain are listed as tasks in your plan. Work against any task in your plan is timed.

- When you begin work on a task, you start the timer in the tool.
- When you stop work on a task, you stop the timer.
- The tool calculates durations automatically.
- If your task is interrupted, you can stop and then restart your timer to resume.
- If you forget to use your timer, you estimate your time-on-task, and enter it manually.

How Am I Spending My Time?



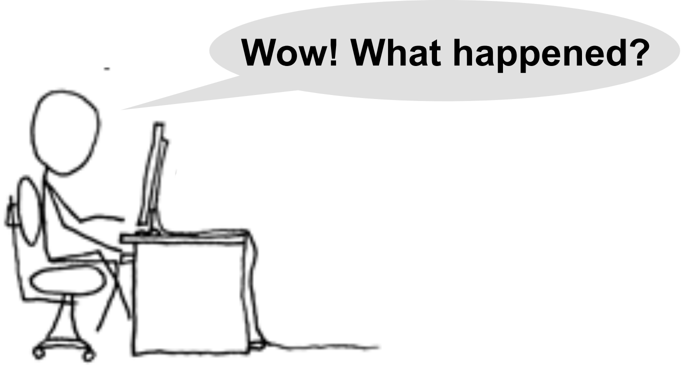
Time

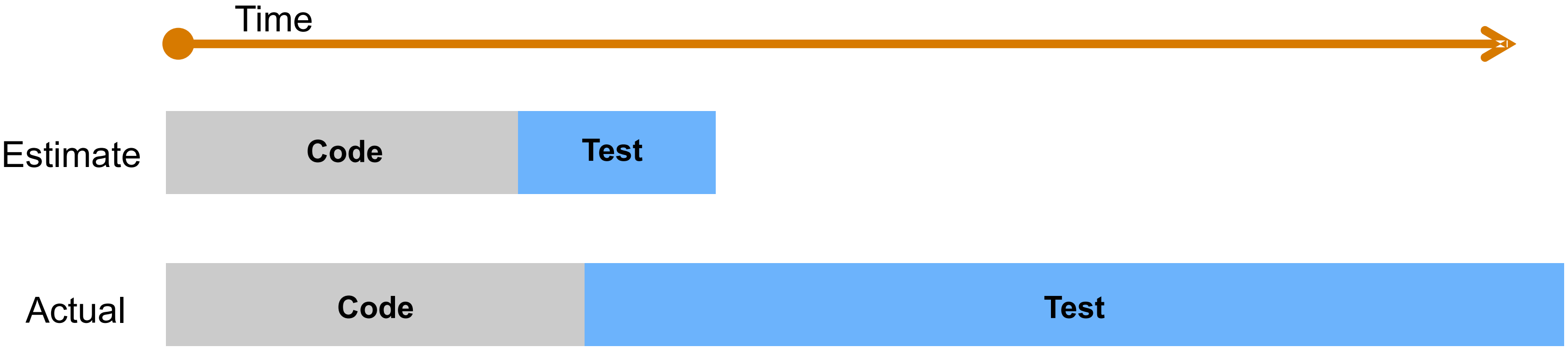
Estimate

Code

Test

How Am I Spending My Time?





Time

Estimate

Code

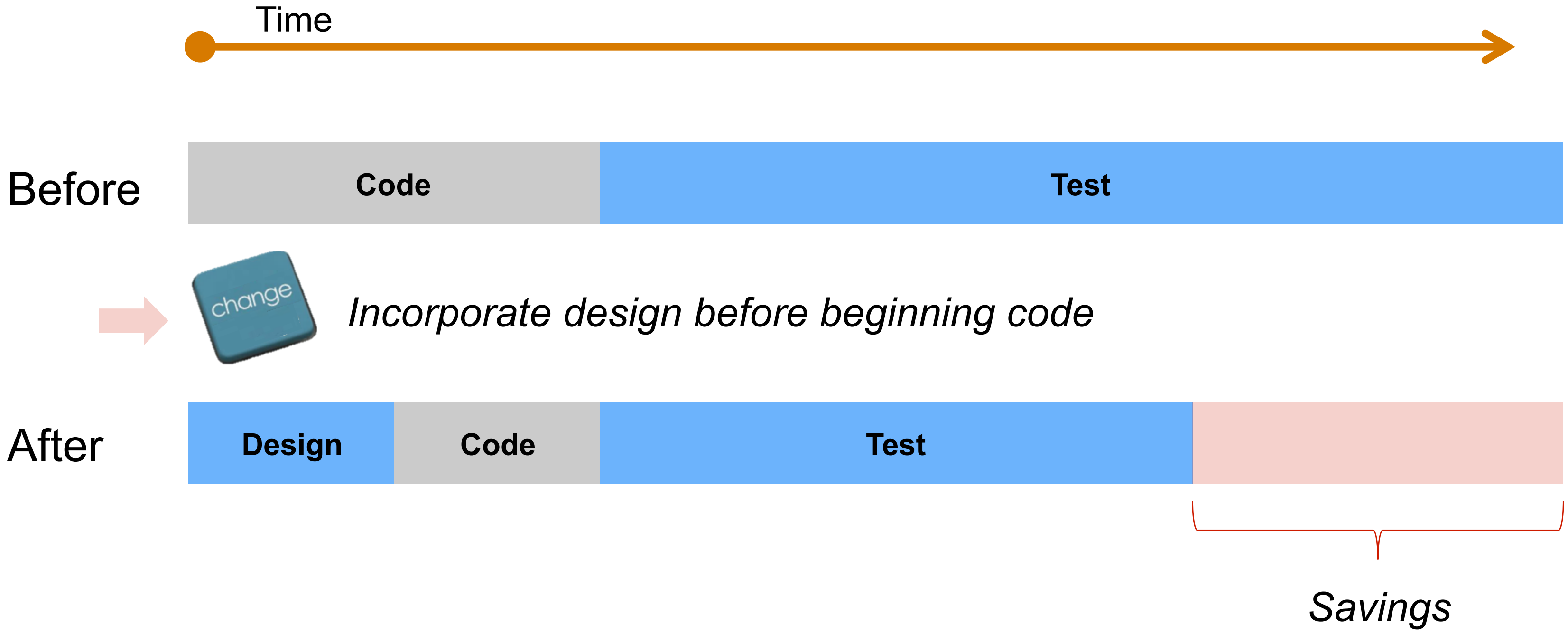
Test

Actual

Code

Test

How Am I Spending My Time?

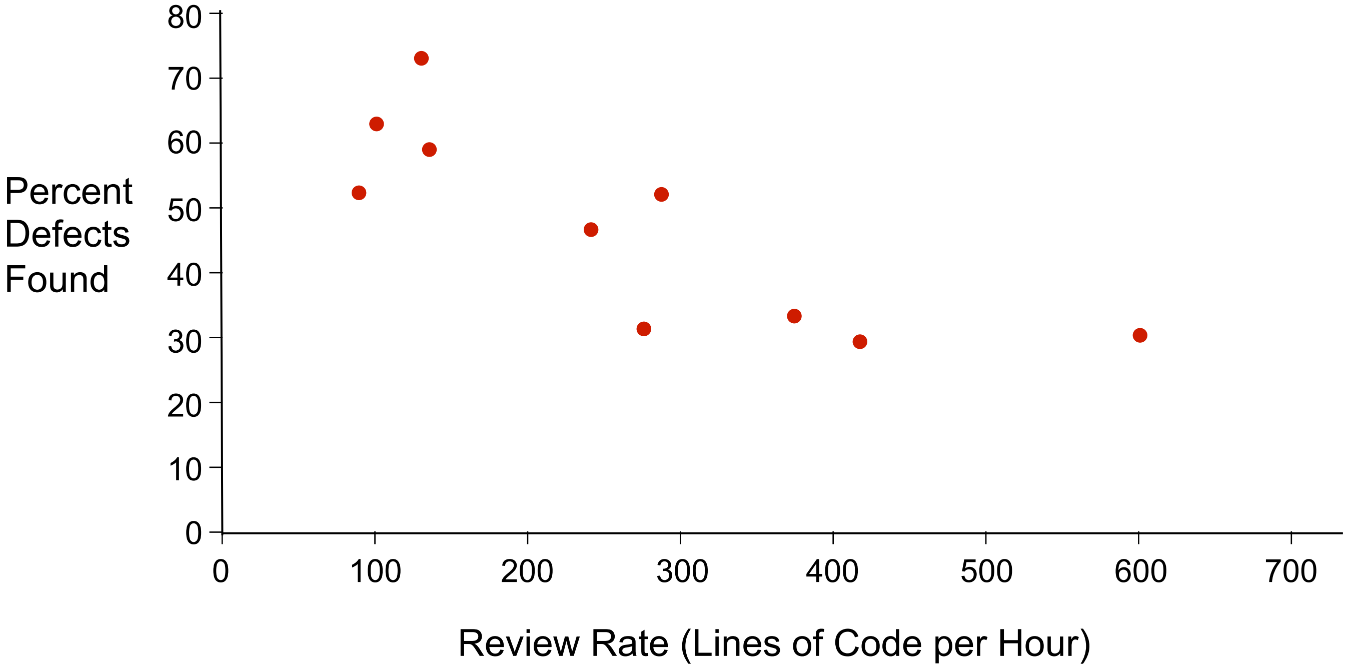


Incorporate design before beginning code

Savings

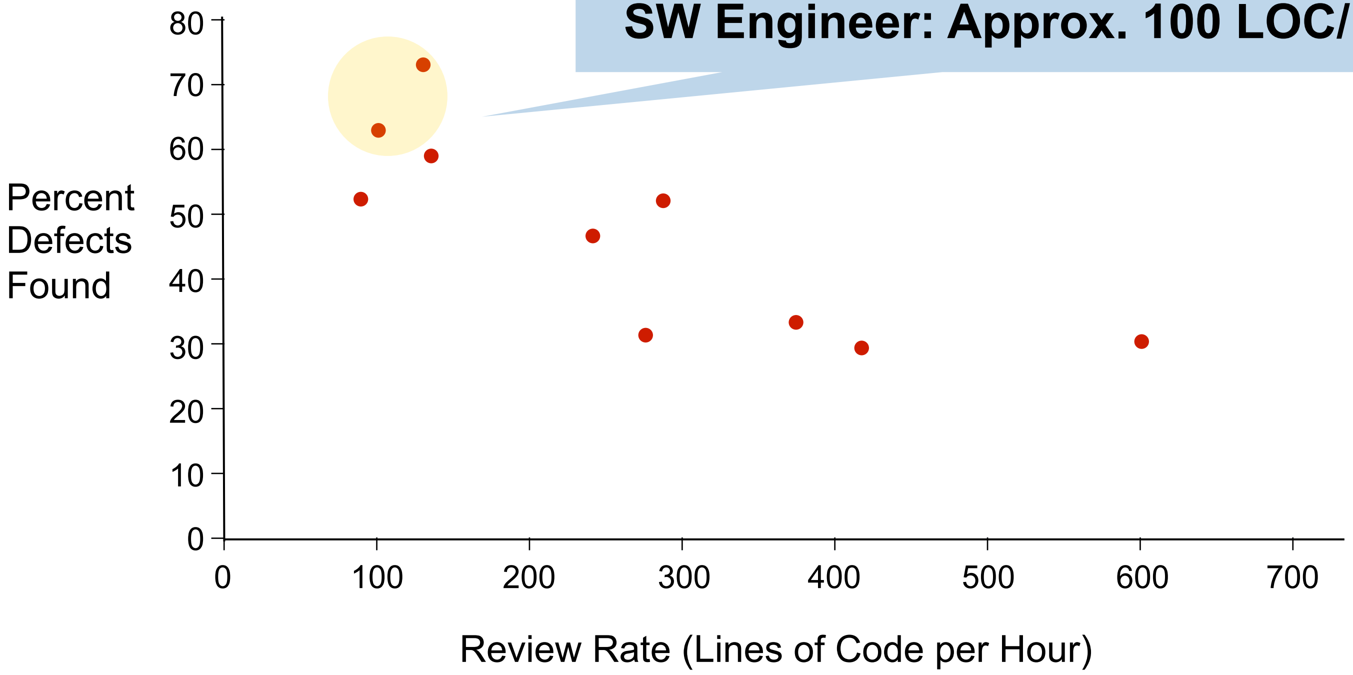
Time is Money

Analysis of Data to Improve



[Humphrey 2005, page 192]

Analysis of Data to Improve

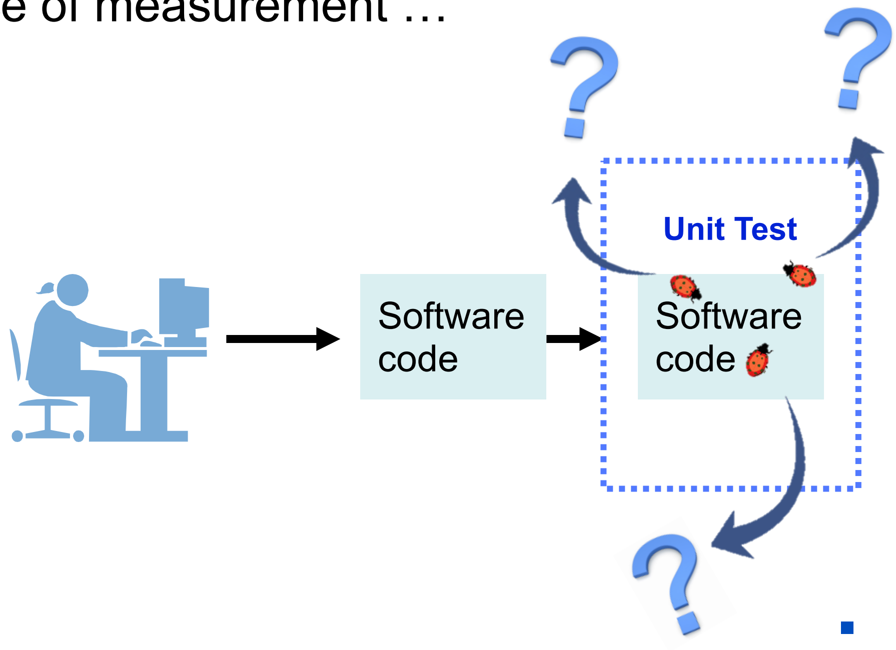


Optimal review rate for this SW Engineer: Approx. 100 LOC/Hr.

[Humphrey 2005, page 192]

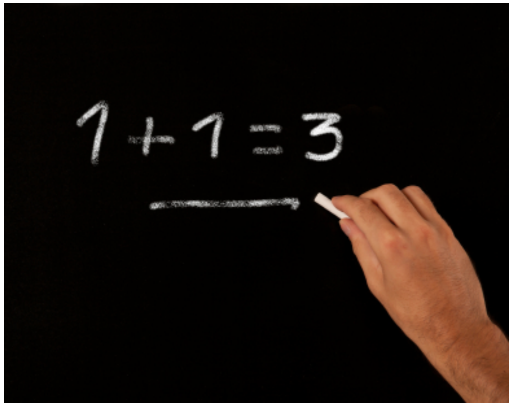
What About Quality Performance?

In the absence of measurement …



- No feedback loop
- No learning
- Mistakes will happen again

Defect Tracking

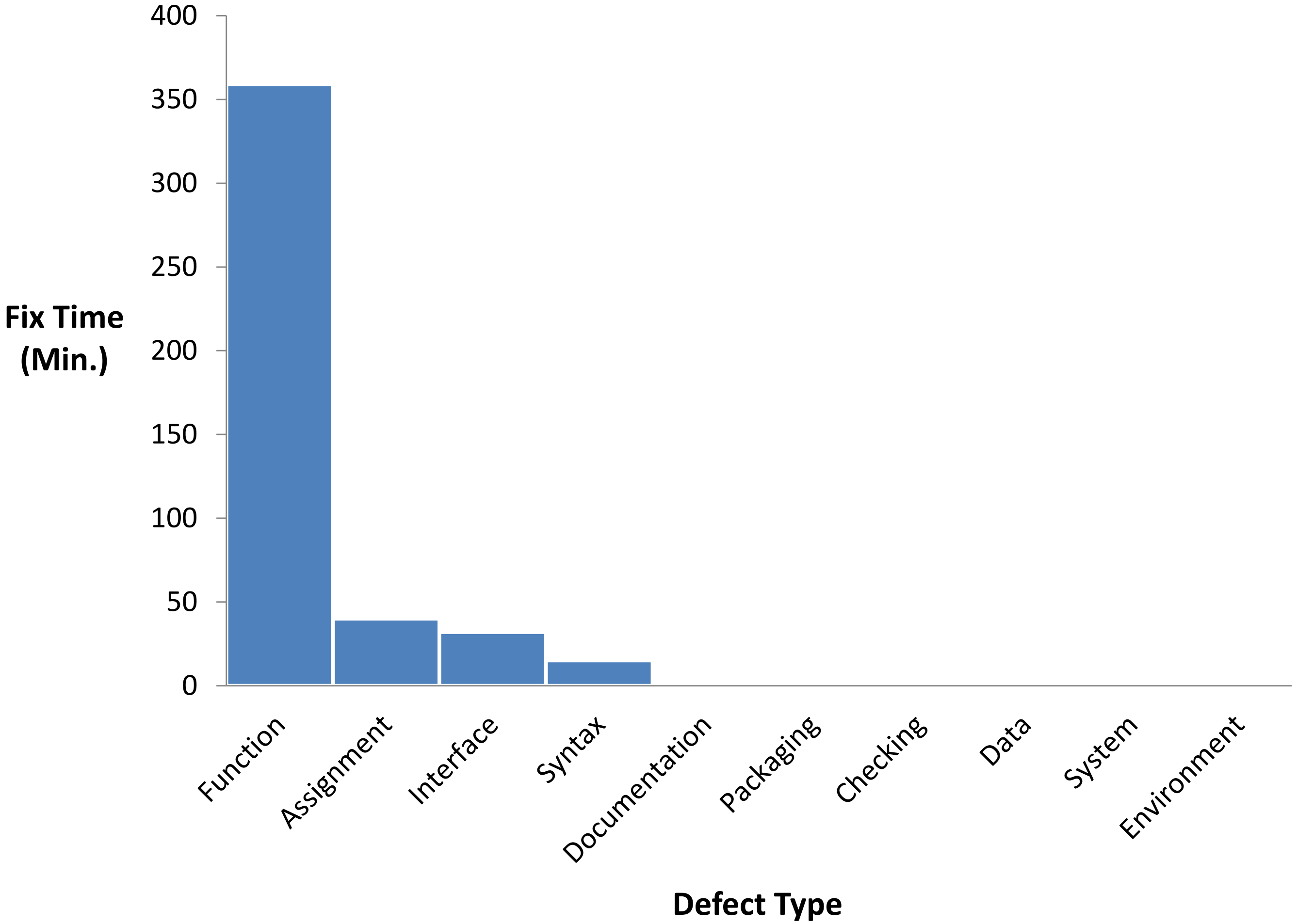


Most defects are discovered during personal reviews, inspections, and other quality control activities.

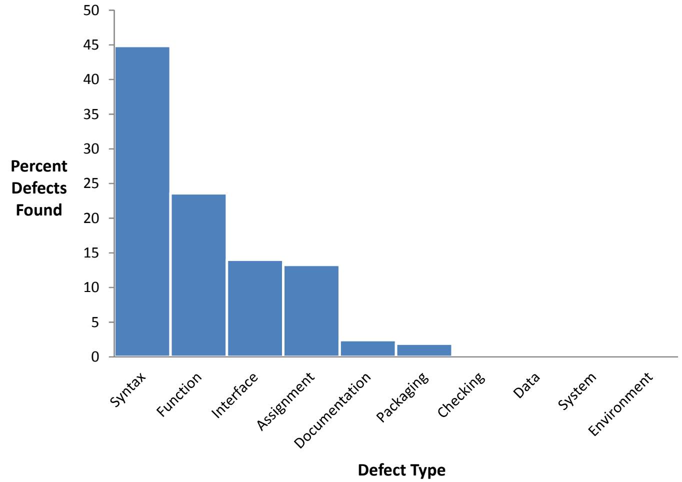
Whenever a defect is found, you open the defect log of the planning/tracking tool and record the following:

- the start time when defect was found
- defect type
- the process phase where the defect was injected
- the process phase where the defect was removed
- a brief description of the defect
- the stop time (when you have completed *fixing* the defect)

Fix Time by Defect Type



Improving Review Practices



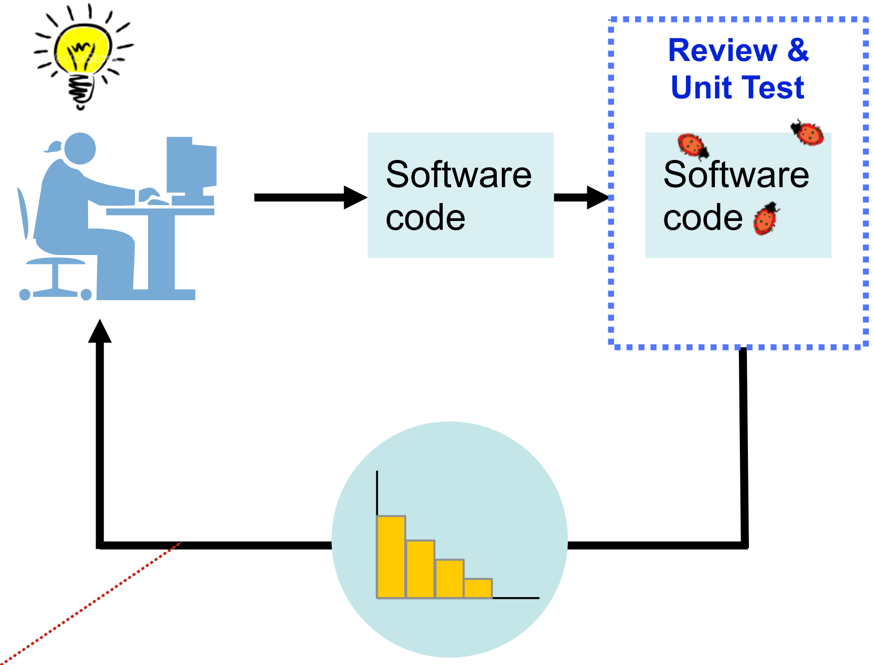
Use data to update review checklist.

Prevent errors from occurring again and again.

```
      <div id="copySpaceBg"></div>
      <div id="copySpaceGrid">
        <div class="csNW"></div>
        <div class="csN"></div>
        <div class="csNE"></div>

        <div class="csW"></div>
        <div class="csCenter"></div>
        <div class="csE"></div>
        <div class="csSW"></div>
        <div class="csS"></div>
        <div class="csSE"></div>
      </div>
    </div>
    <p>Click where your text will appear.</p>

    <p><a href="#">Apply</a>
    <br class="clear" />
  </div>

  <div id="largePhotoSelector" class="subFilterListNoBorder">
    <div id="largePhoto_list">
      <div><label>L<br /><input type="checkbox" /></label></div>
      <div><label>M<br /><input type="checkbox" /></label></div>

      <div><label>S<br /><input type="checkbox" /></label></div>
      <br class="clear" />
    </div>
  </div>
</div>

<div id="illustrationsComplexitySelector" class="subFilterNoBorder">
    <div id="illustration_header" class="subFilter">Illustration</div>

  <div id="illustration_list">
```

Using Measurement To Understand …

… and, to get better



Closed-loop feedback & learning

- Defects recorded by type and time-to-fix
- Closed-loop feedback
- Learn from mistakes so they don't happen again

Taking Responsibility

Others cannot manage how you estimate your work and how you manage the quality of our work.

Knowledge workers manage themselves with data.

Software engineers are knowledge workers.

Only four basic measures are need to manage your work.

Can software engineers leverage goal-setting and measurement the way that star athletes do?

Yes! Absolutely.

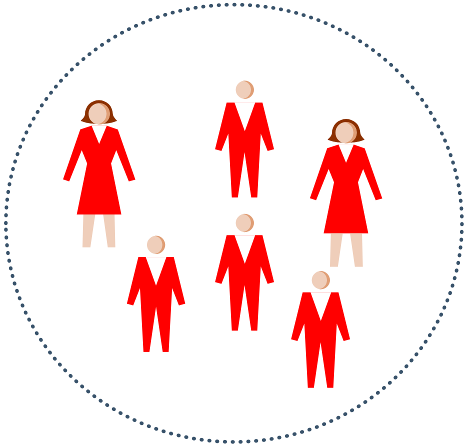
When Measurement Benefits the Measured
by Mark Kasunic and William Nichols
04.23.14 • 1:30 pm ET–2:30 pm ET

William Nichols
Senior Member of the Technical Staff
Software Engineering Institute

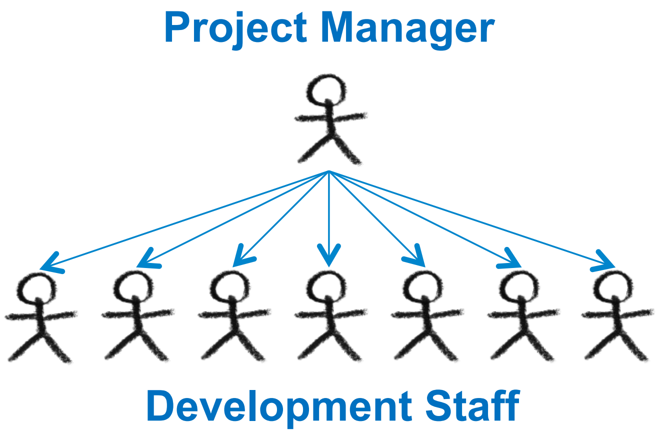
Bill Nichols joined the Software Engineering Institute (SEI) in 2006 as a senior member of the technical staff and serves as a Personal Software Process (PSP) instructor and Team Software Process (TSP) Mentor Coach with the TSP Initiative within the Software Solutions Division (SSD). Prior to joining the SEI, Dr. Nichols lead a software development team at the Bettis Laboratory near Pittsburgh, Pennsylvania, where he had been developing and maintaining nuclear engineering and scientific software for 14 years. His TSP publications include the the PSP and TSP Bodies of Knowledge, The TSP Coach Mentoring Program Guidebook, and various publications addressing software quality planning. Research publications include an algorithm for use in neutron diffusion programs, design and performance of a physics data acquisition system, and experimental results in particle physics. He has a doctorate in physics from Carnegie Mellon University.

Measurement On Your Team

The Team

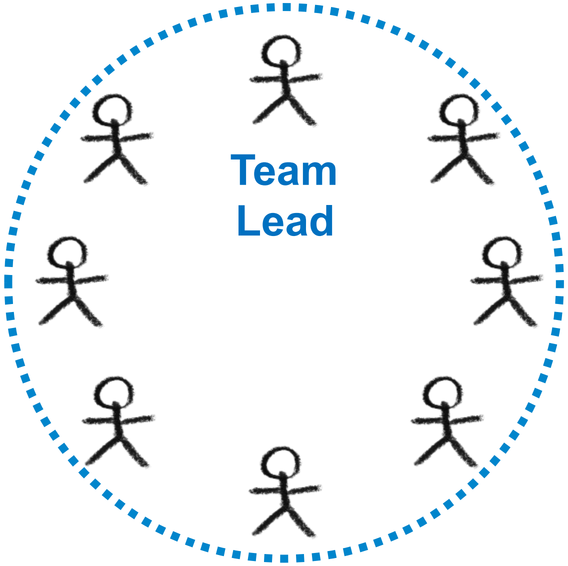


Self-Managed Team of Knowledge Workers



Project Manager

Development Staff



Team Lead

Traditional Project

The PM plans, directs, and tracks the work.

Self-Managed Team

The Team directs, and tracks the work.

Self-Managed Teams Plan Their Work

Management provides the goals and constraints for the project.

The team then develops its plan for meeting management's objectives.

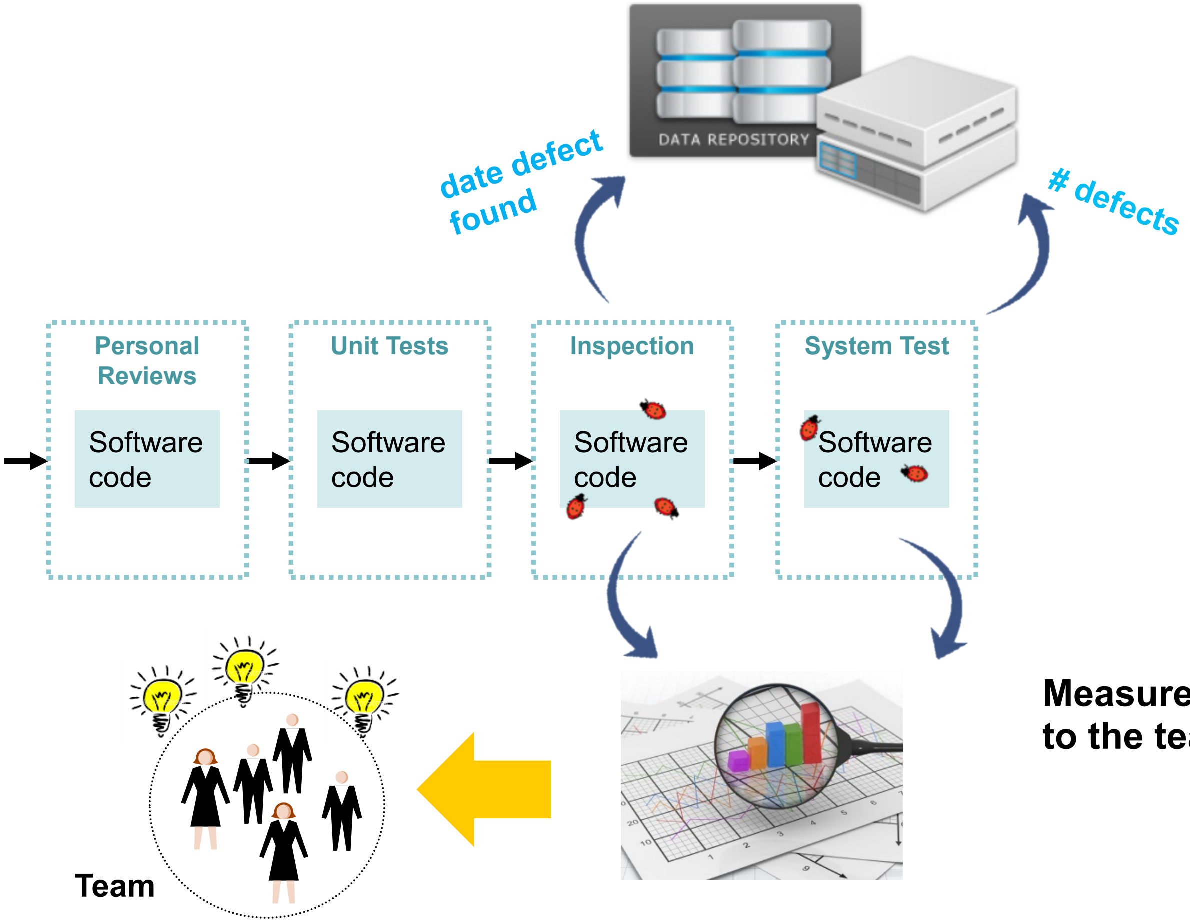
If necessary, the team negotiates with management to arrive at a mutually agreeable outcome.

But it's up to the team to manage their work! Not someone else.

The team must have data to manage the work, to meet their commitments.

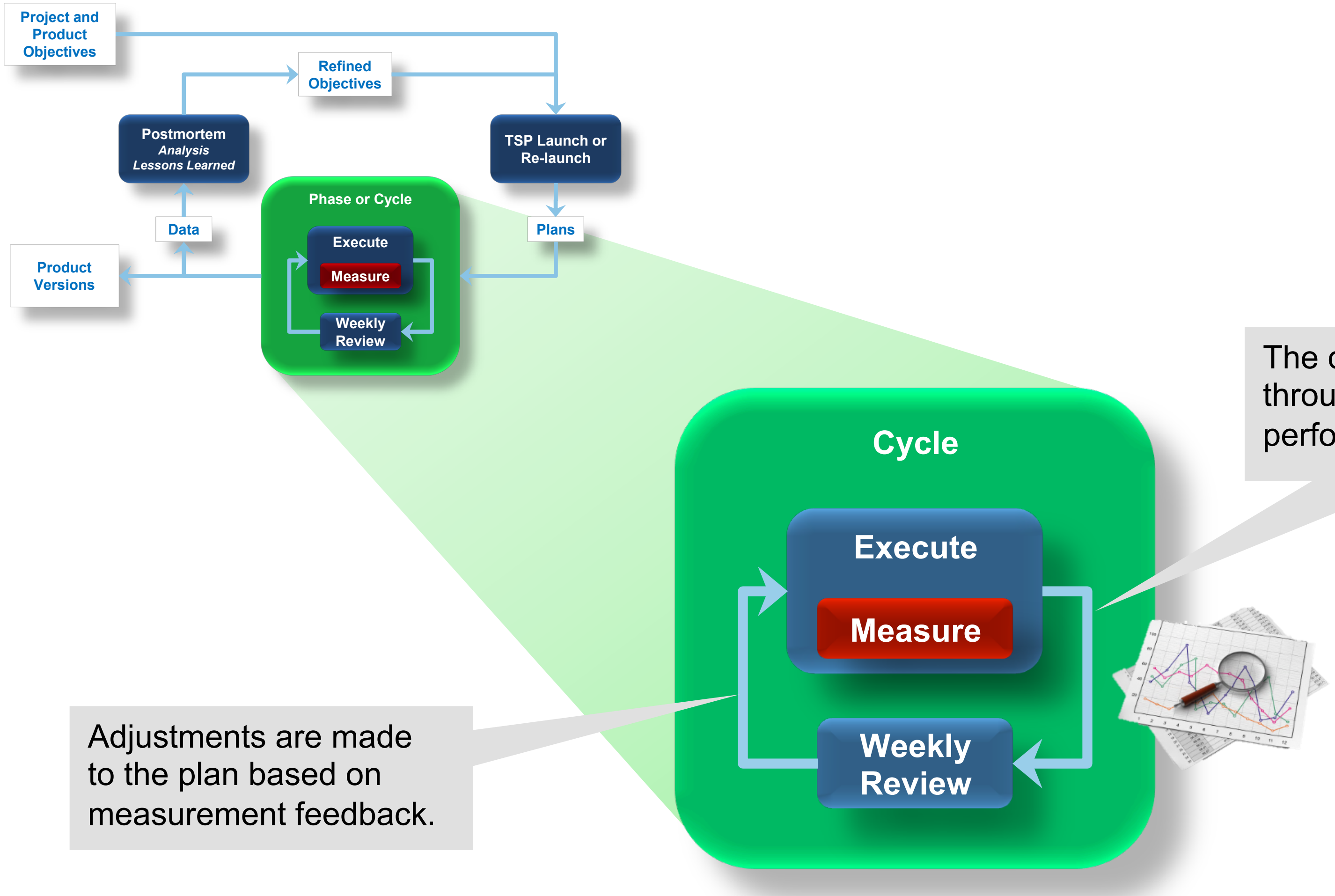
Measurement helps teams manage their commitments.

Measurement To Benefit the Measured



Measurement feedback to the team

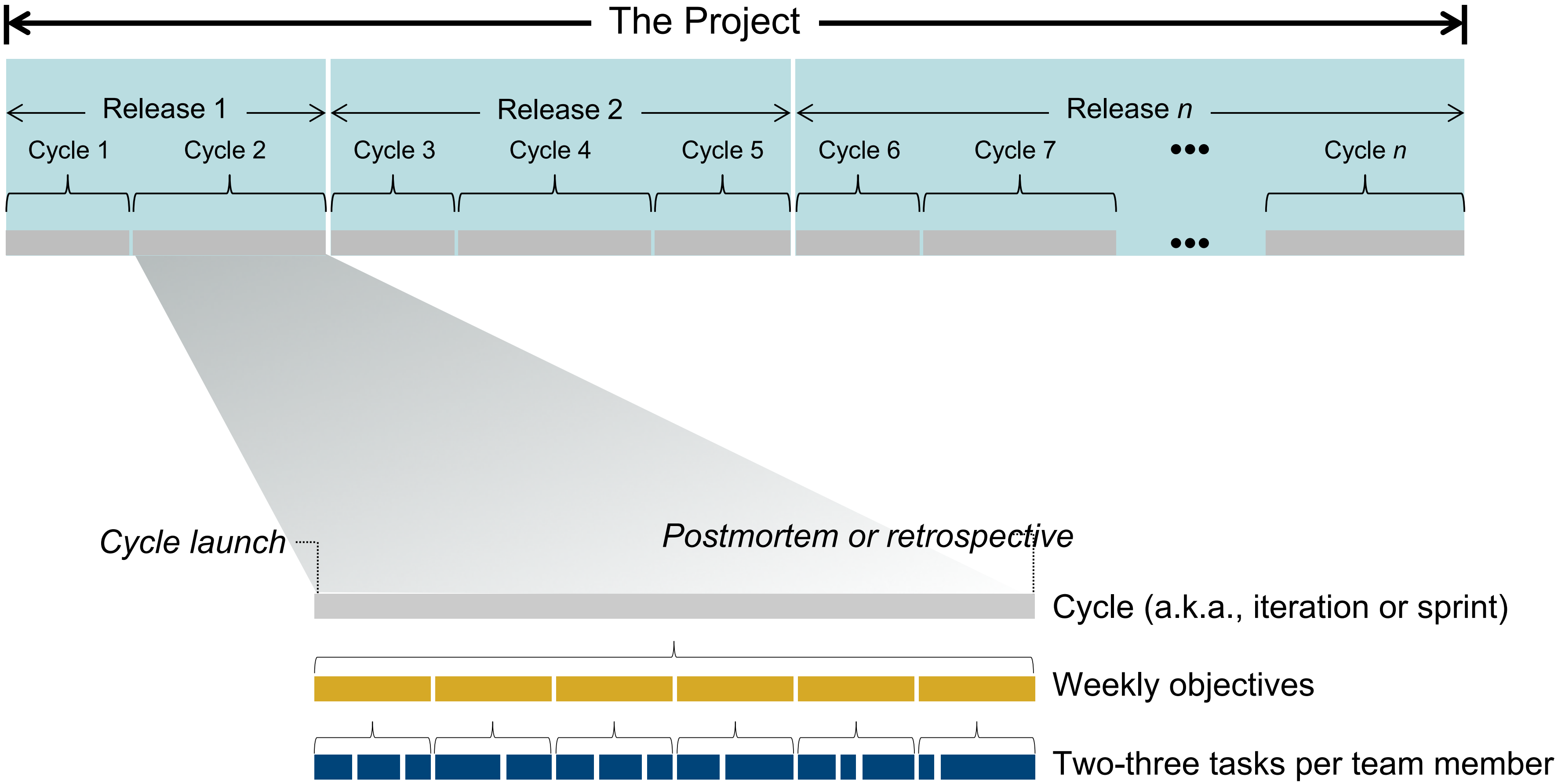
Measurement Used to Manage



The data is used throughout to assess performance.

Adjustments are made to the plan based on measurement feedback.

Planning, Doing, and Learning



Comparing Estimates to Actuals

For both schedule and quality …

Is the project on track?

Comparing Estimates to Actuals

Closed loop feedback

For both schedule and quality …

Is the project on track?

Comparing Estimates to Actuals

For both schedule and quality …

Closed loop feedback

Learning

Is the project on track?

Comparing Estimates to Actuals

For both schedule and quality …

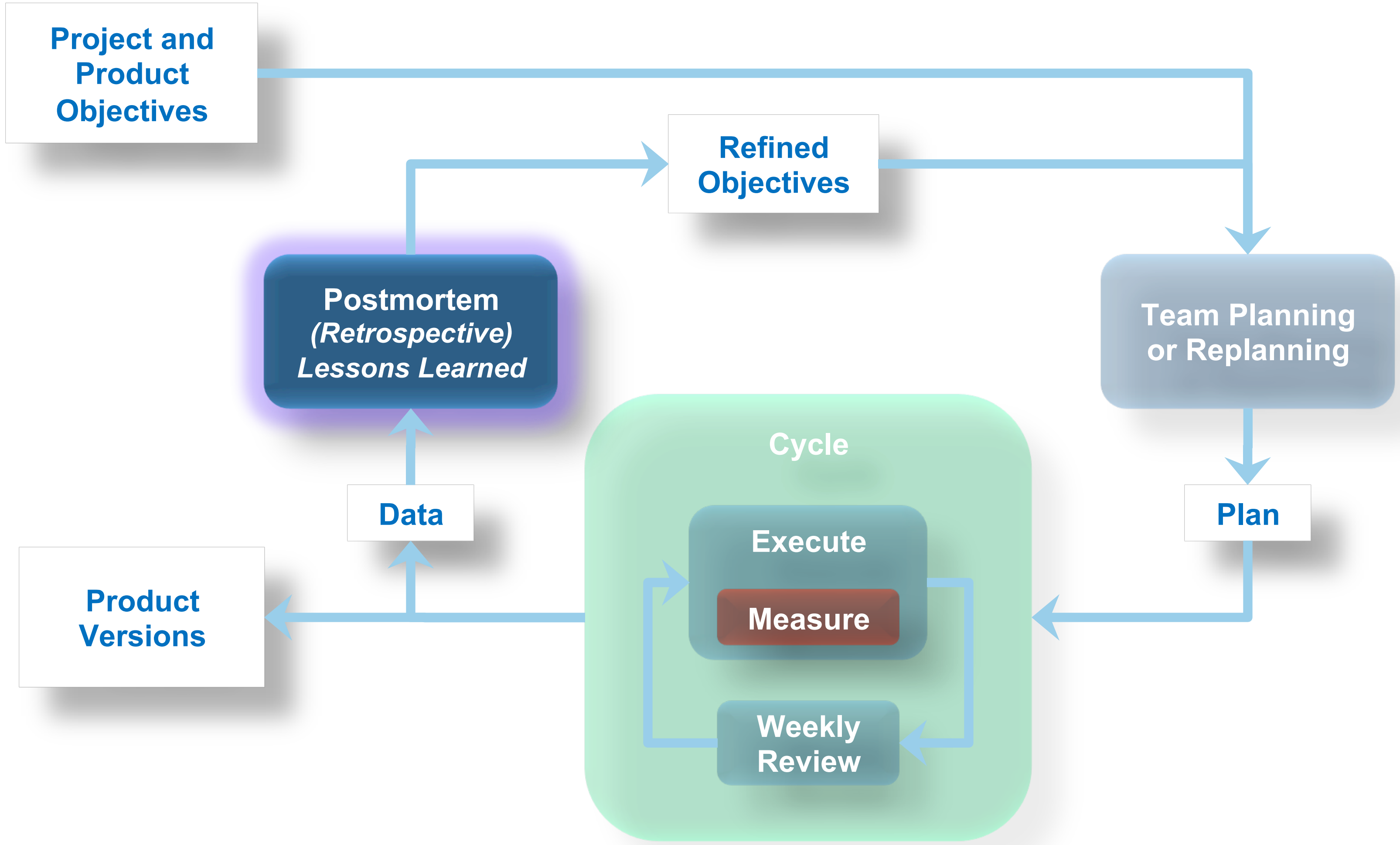
Closed loop feedback

Learning

Performance improvement

Is the project on track?

Measurement That Benefits the Measured



Again … Only Four Basic Measures Needed

Software engineers only need to collect four basic measures to mange their schedule performance and the quality of their work.

Time on Task

Size

Defects

Schedule

Derived From the Four Basic Measures

Many other useful measures and indicators can be easily derived from the four basic measures including:

- estimation accuracy*
- prediction intervals*
- time in phase distribution
- defect injection distribution
- defect removal distribution
- productivity
- reuse percentage

- cost performance index
- planned value
- earned value
- predicted earned value
- defect density
- defect density by phase
- defect removal rate by phase

- defect removal leverage
- review rates
- process yield
- phase yield
- failure cost of quality (COQ)
- appraisal COQ
- appraisal/failure COQ ratio

* Both size and time

How Do You Know If It's a *Best Practice?*

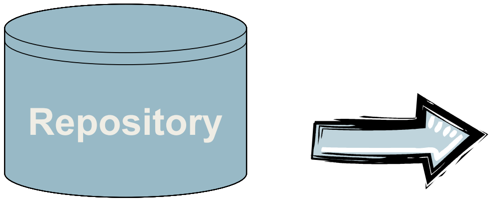
Organizations want a way to gauge their performance and to compare their performance with others in their industry.

Data on project performance is needed to provide evidence of what (exactly) constitutes a best practice.

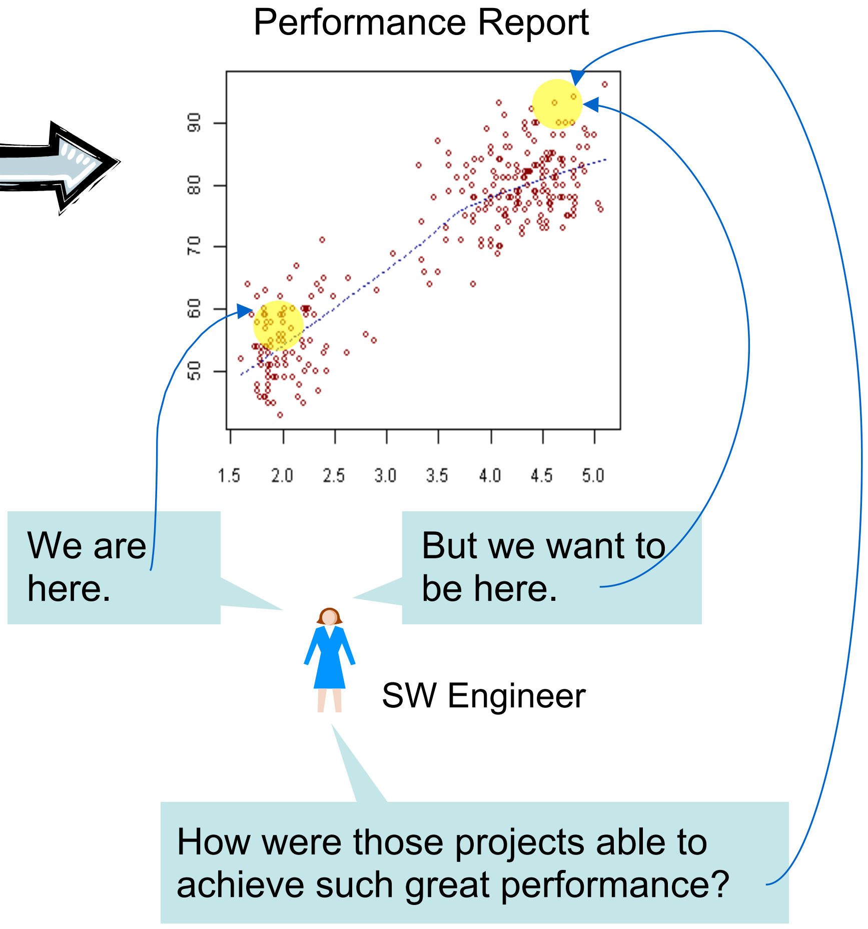
How do you even know what a best practice is unless you measure and compare it to other practices?

Benchmarks provide a reference point for interpreting performance.

Benchmarking & *Best* Practices



You can't benchmark without measurement!



We are here.

But we want to be here.

SW Engineer

How were those projects able to achieve such great performance?

Description of the Data

The data was submitted to the SEI between 2000 and 2012.

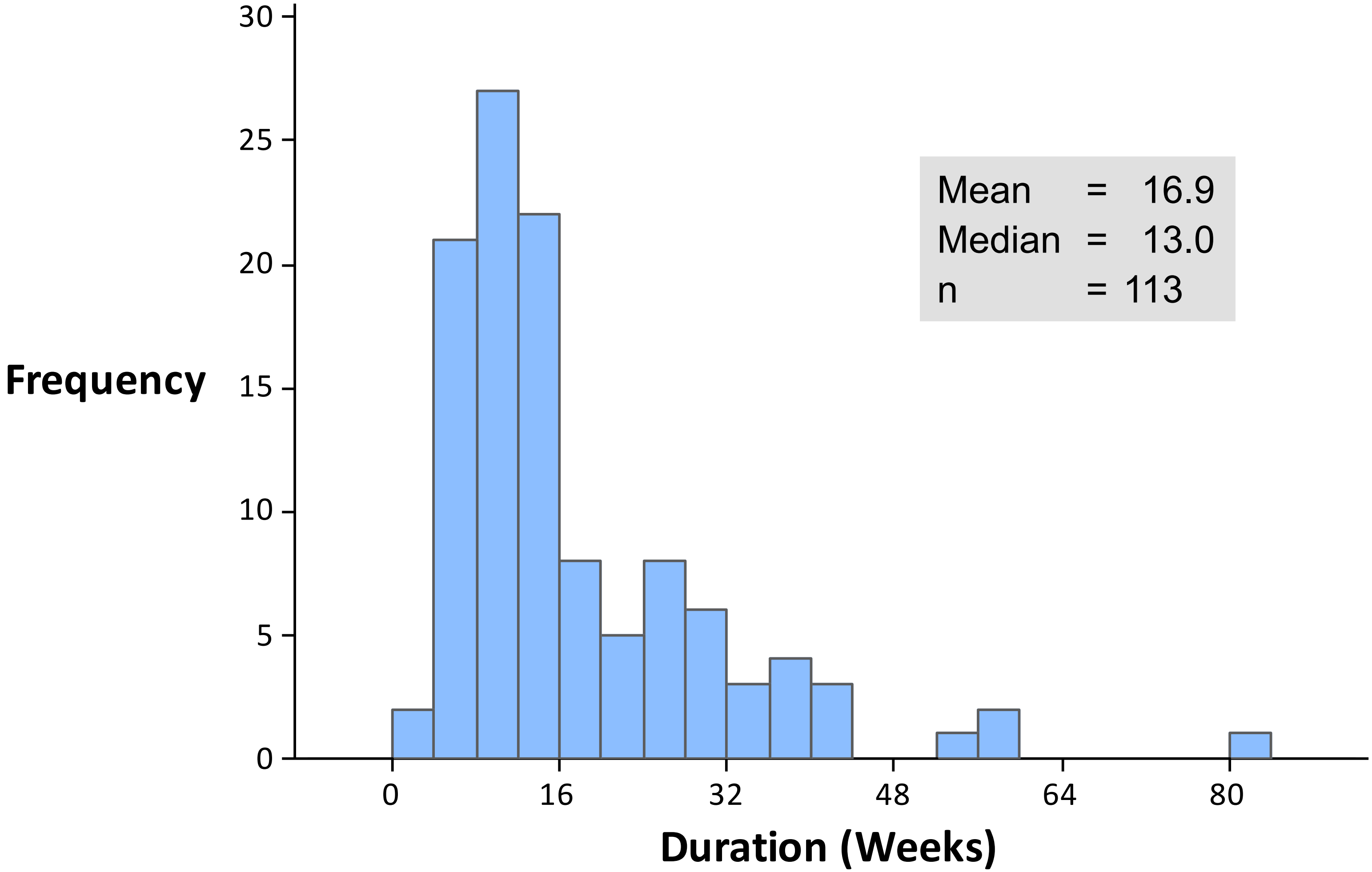
The source data is from 93 projects in the United States and 20 projects from Mexico.

This is data that has been aggregated at the *team* level at the time of a cycle postmortem (retrospective).

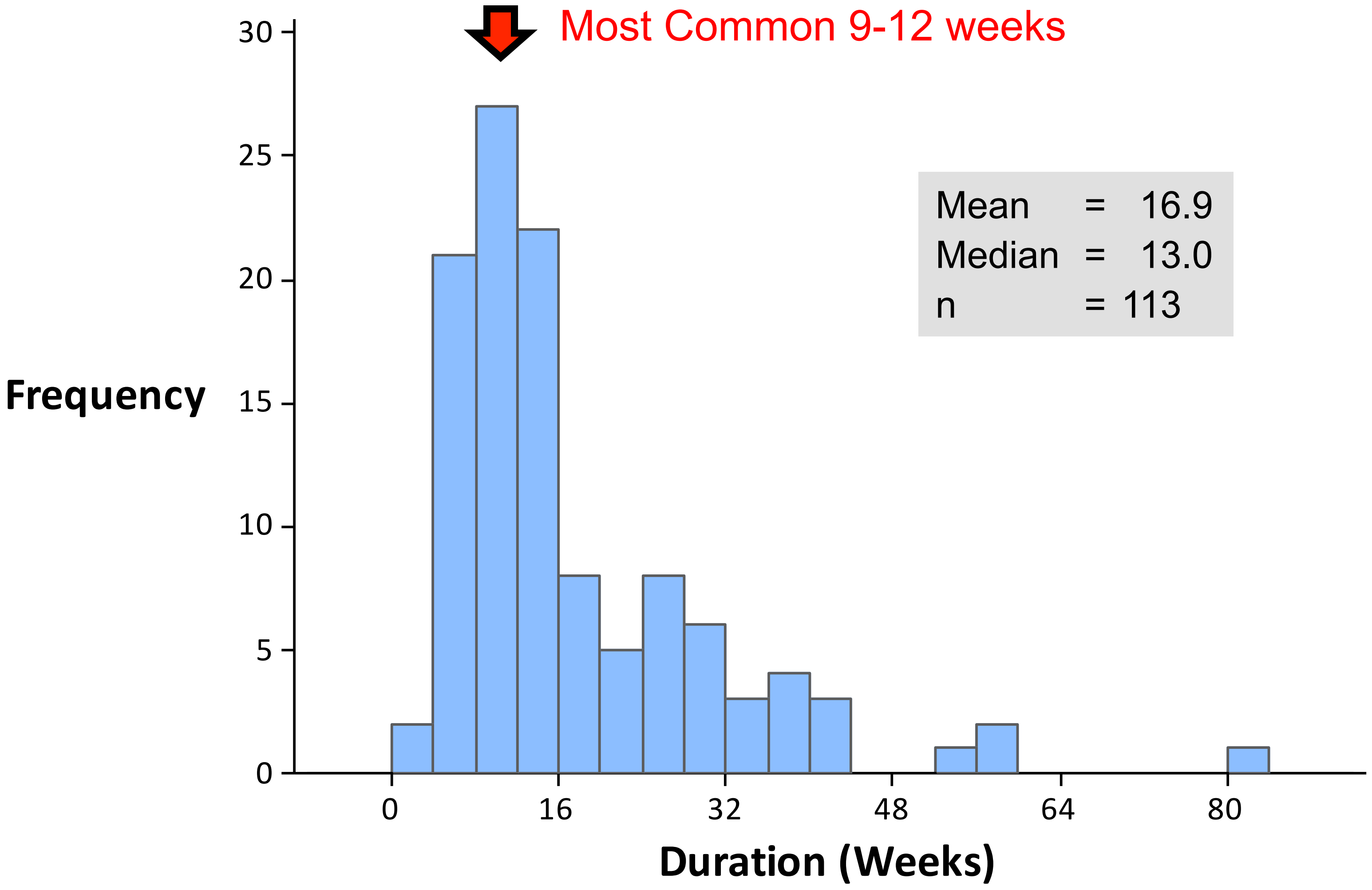
Only data from a project’s last postmortem is included.

Tests were conducted to ensure that extracted data represented unique projects.

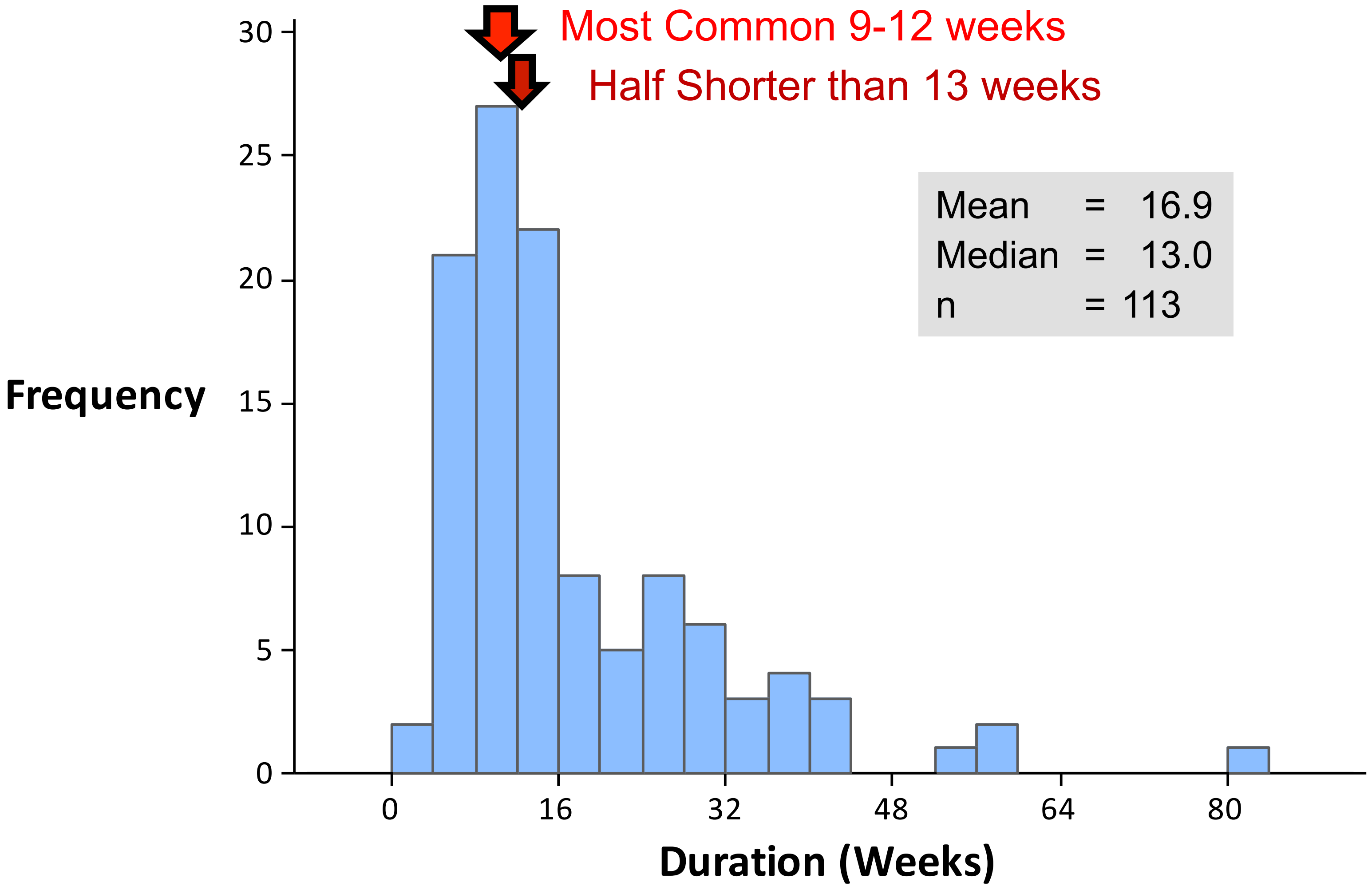
How Long Were Project Durations? [Weeks]



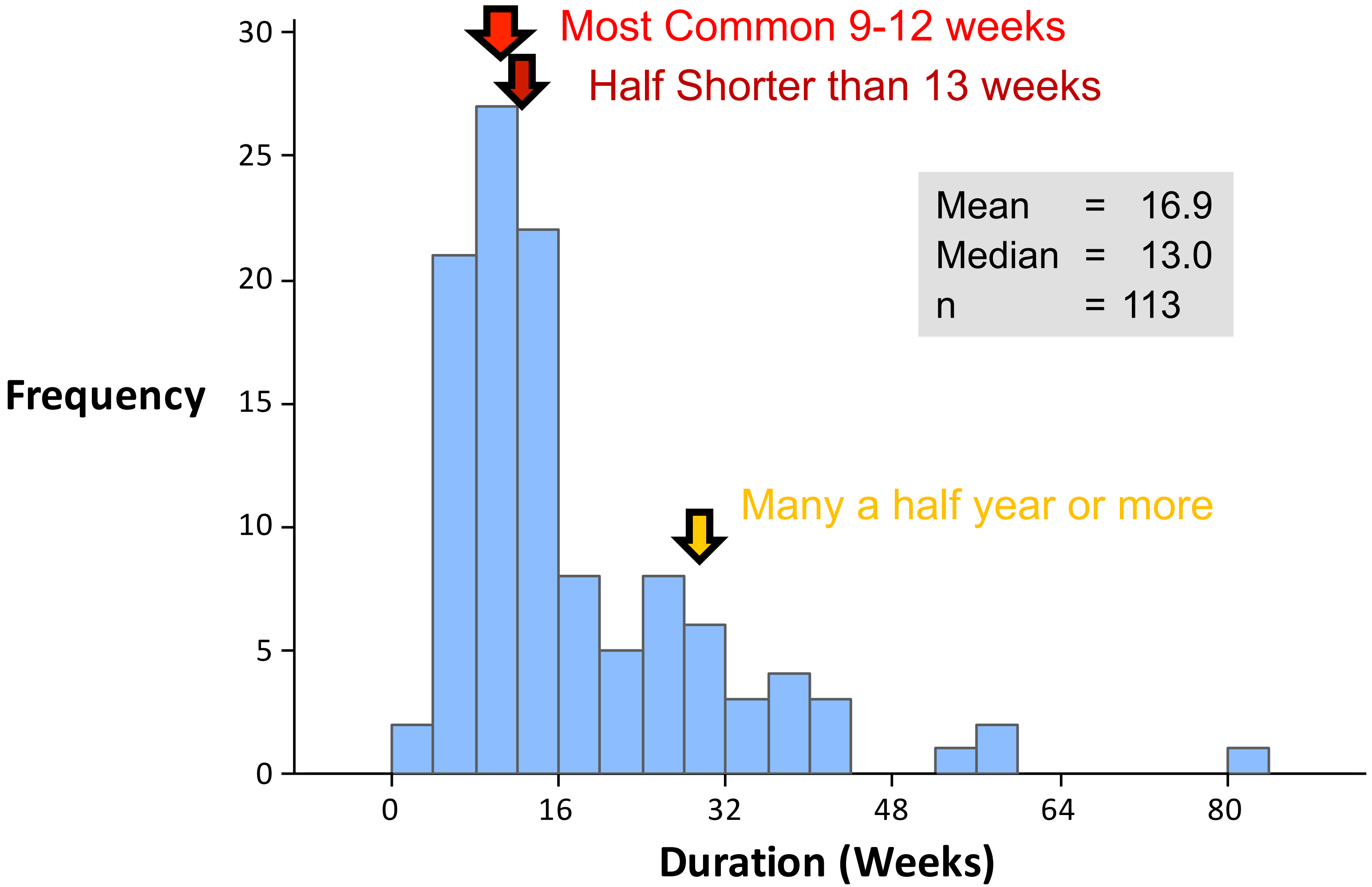
What were Project Durations? [Weeks]



What were Project Durations? [Weeks]



What were Project Durations? [Weeks]



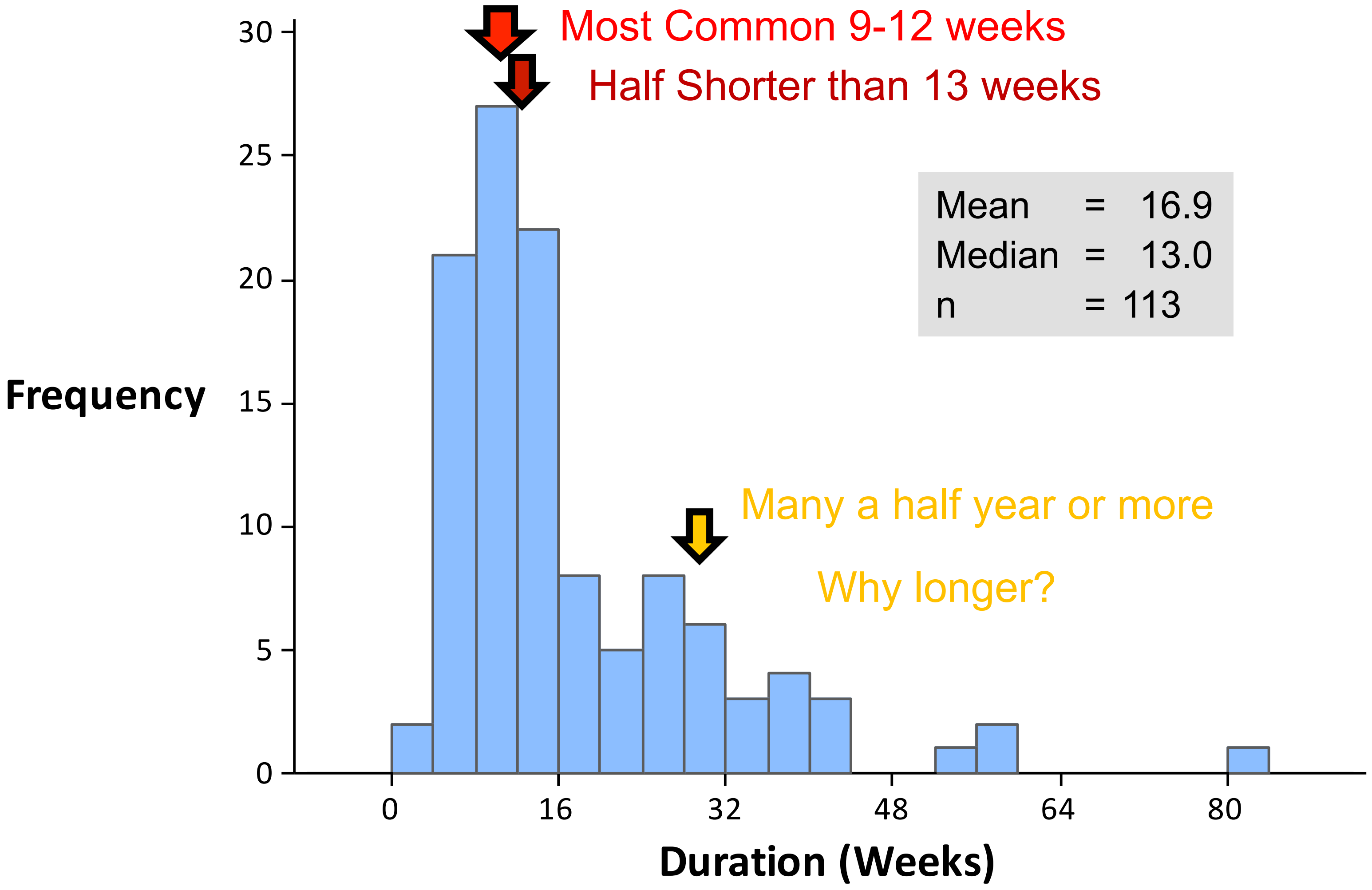
Most Common 9-12 weeks

Half Shorter than 13 weeks

Many a half year or more

Mean = 16.9
Median = 13.0
n = 113

What were Project Durations? [Weeks]



Most Common 9-12 weeks

Half Shorter than 13 weeks

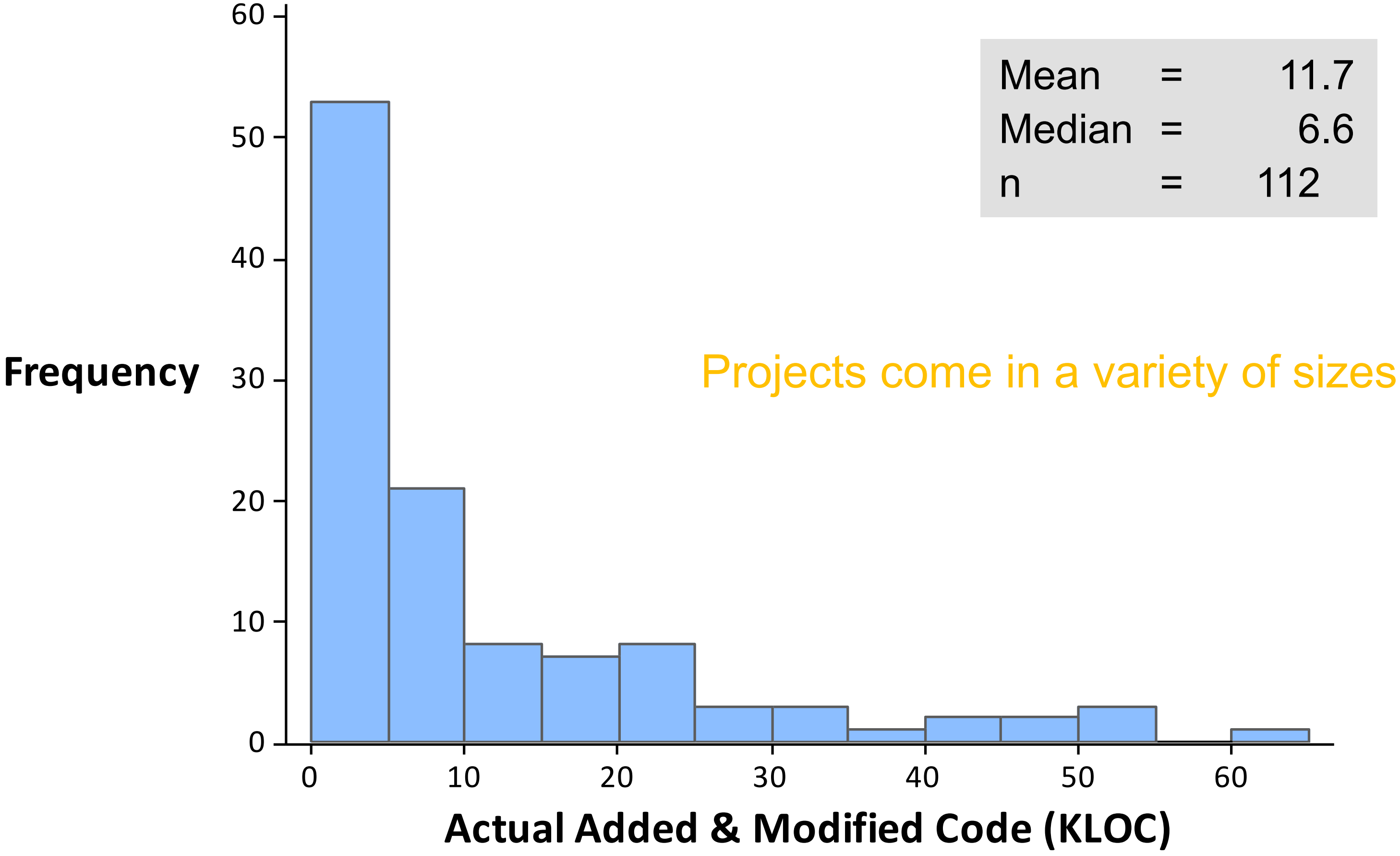
Mean = 16.9
Median = 13.0
n = 113

Many a half year or more

Why longer?

Size - Actual Added and Modified

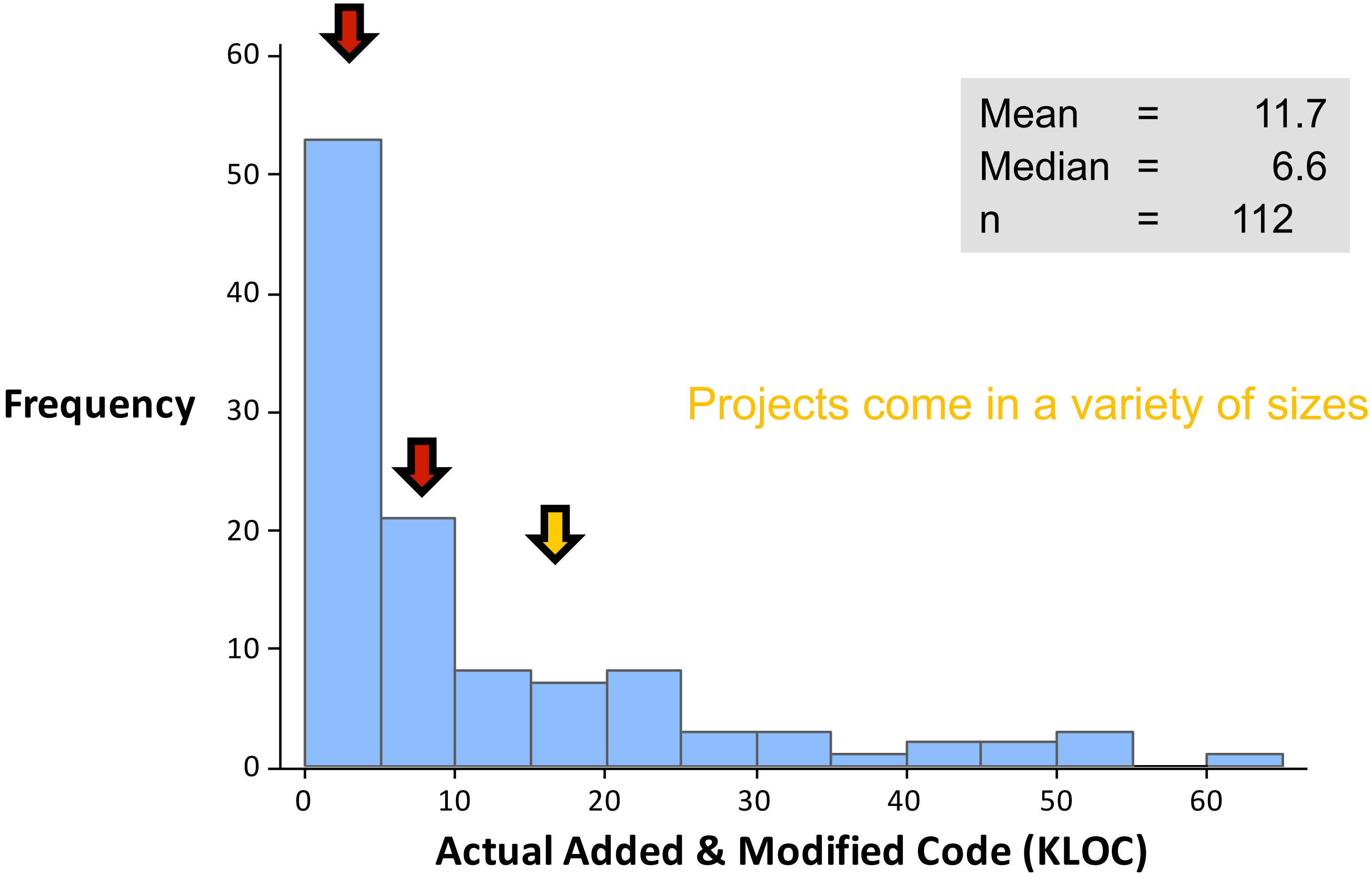
Thousand Source Lines of Code [KLOC]



Projects come in a variety of sizes

Size - Actual Added and Modified

Thousand Source Lines of Code [KLOC]

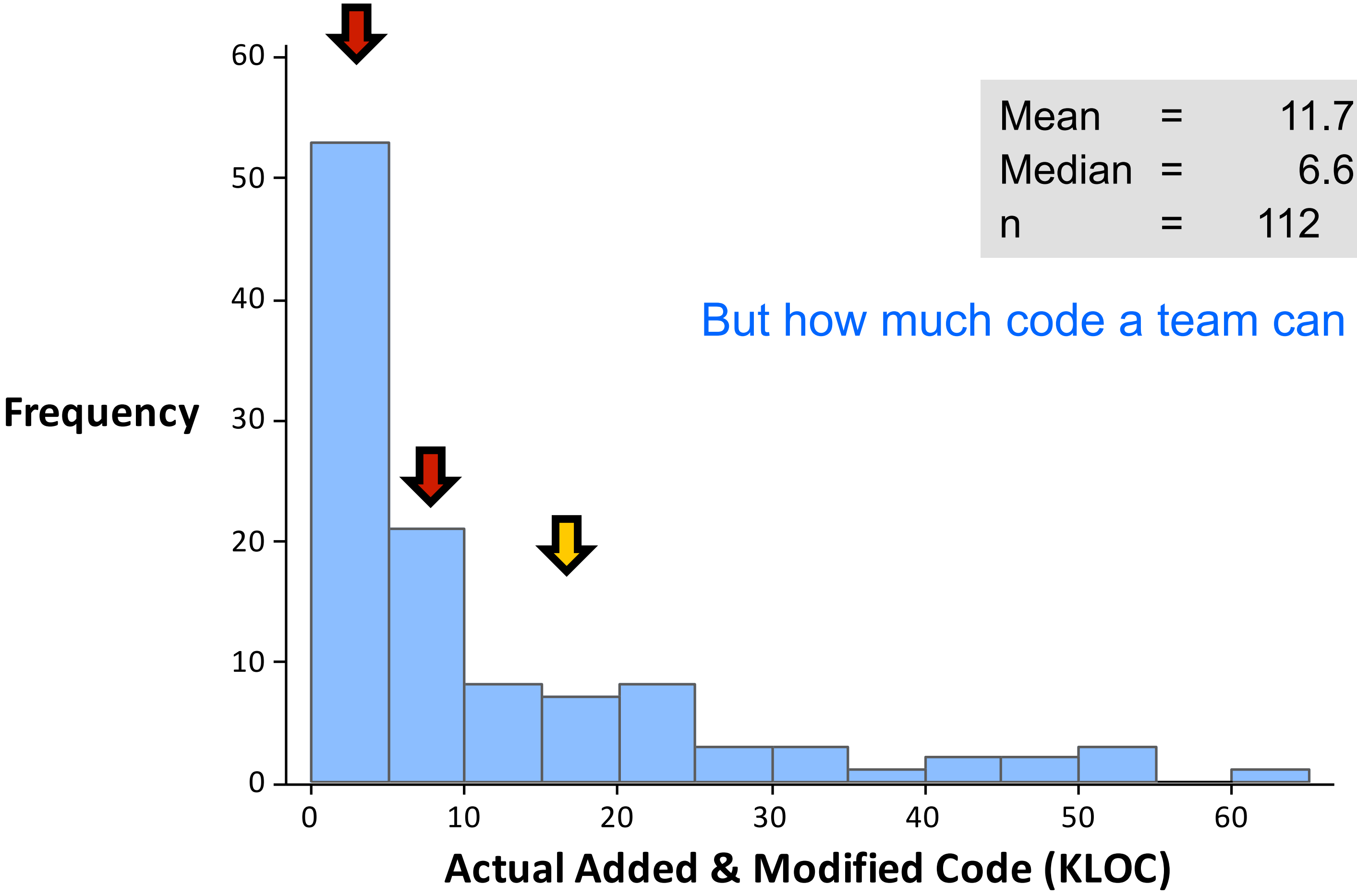


Mean	=	11.7
Median	=	6.6
n	=	112

Projects come in a variety of sizes

Size - Actual Added and Modified

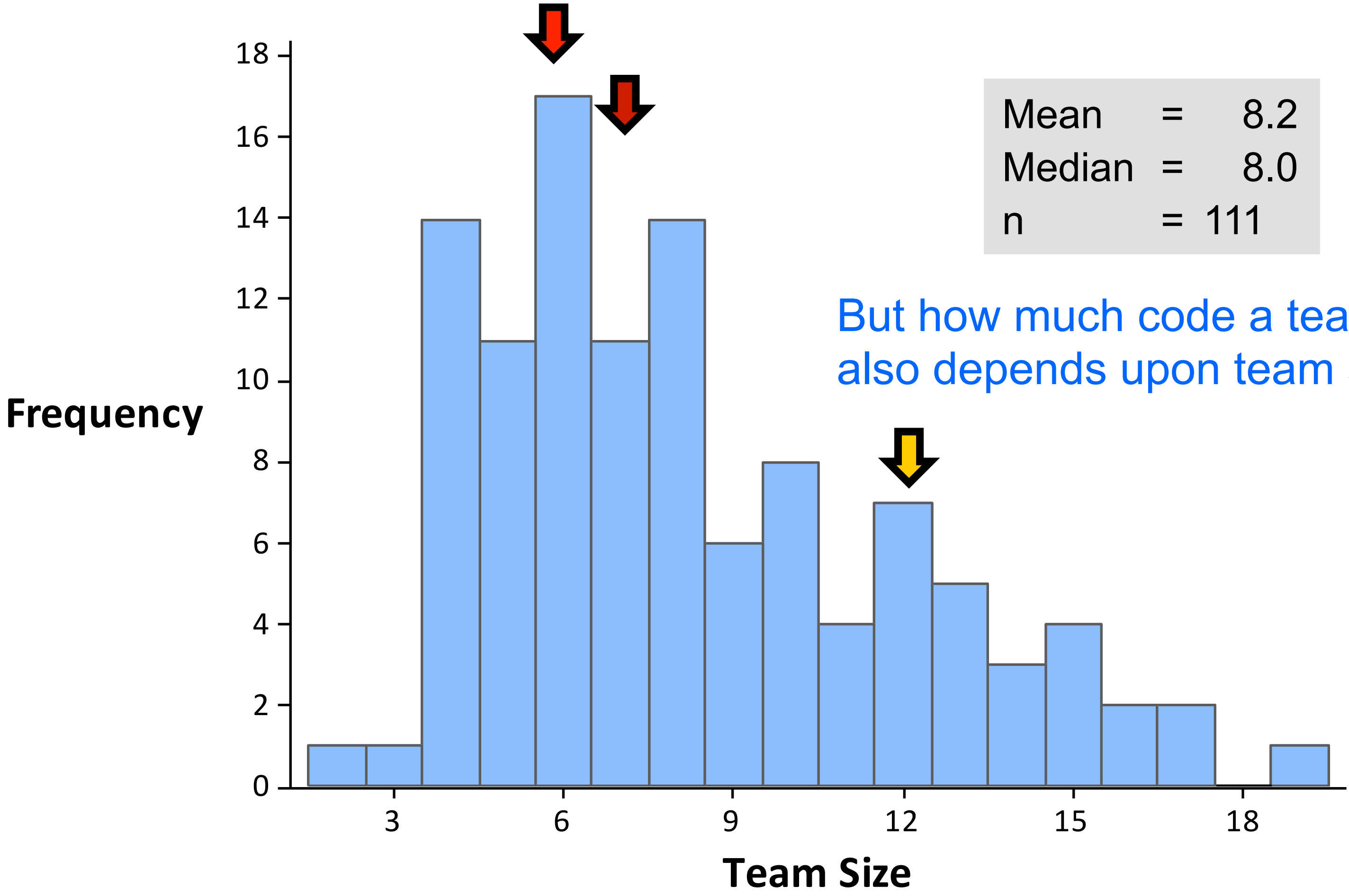
Thousand Source Lines of Code [KLOC]



Mean	=	11.7
Median	=	6.6
n	=	112

But how much code a team can write

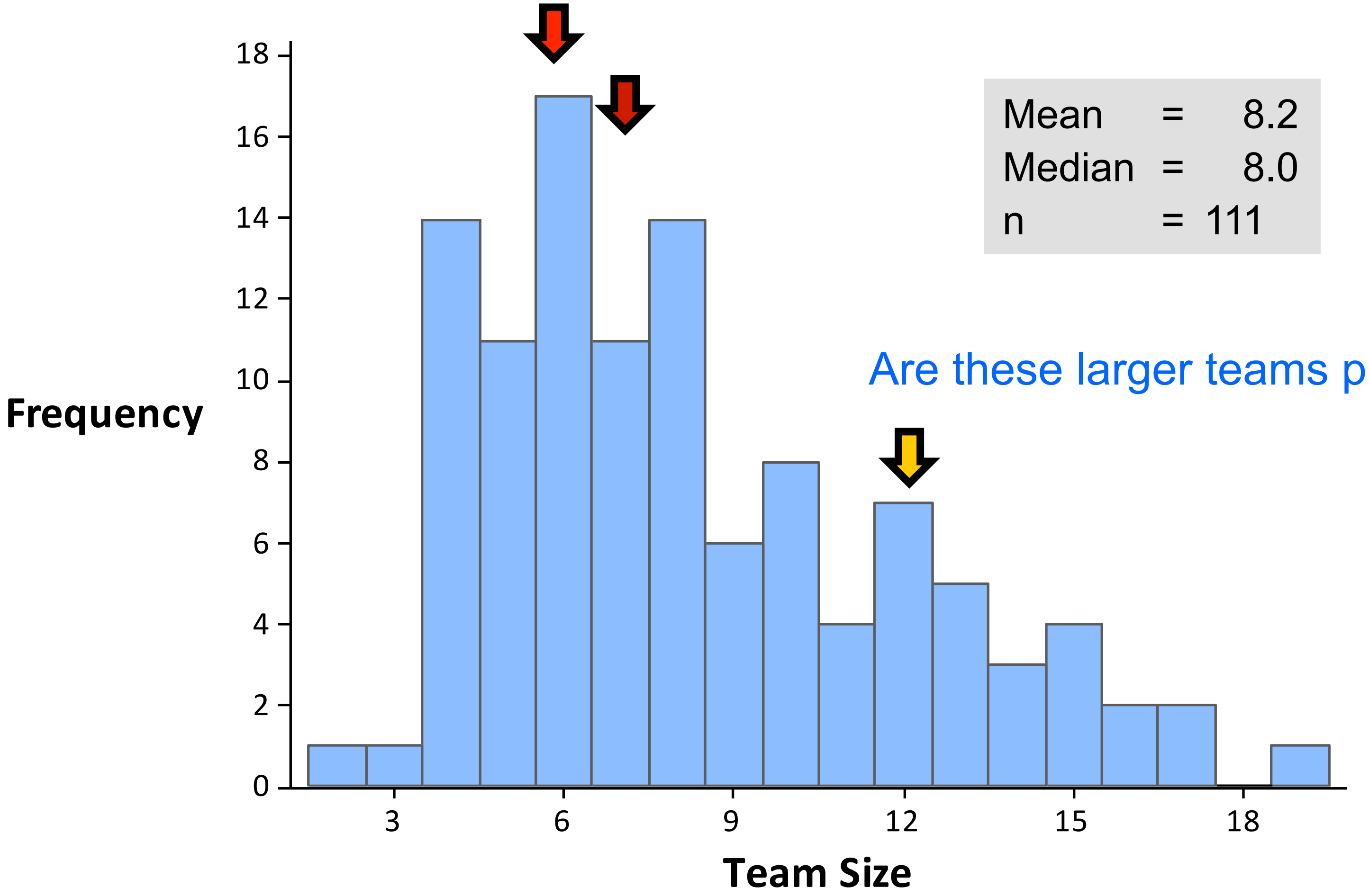
Team Size



Mean	=	8.2
Median	=	8.0
n	=	111

But how much code a team can write also depends upon team size.

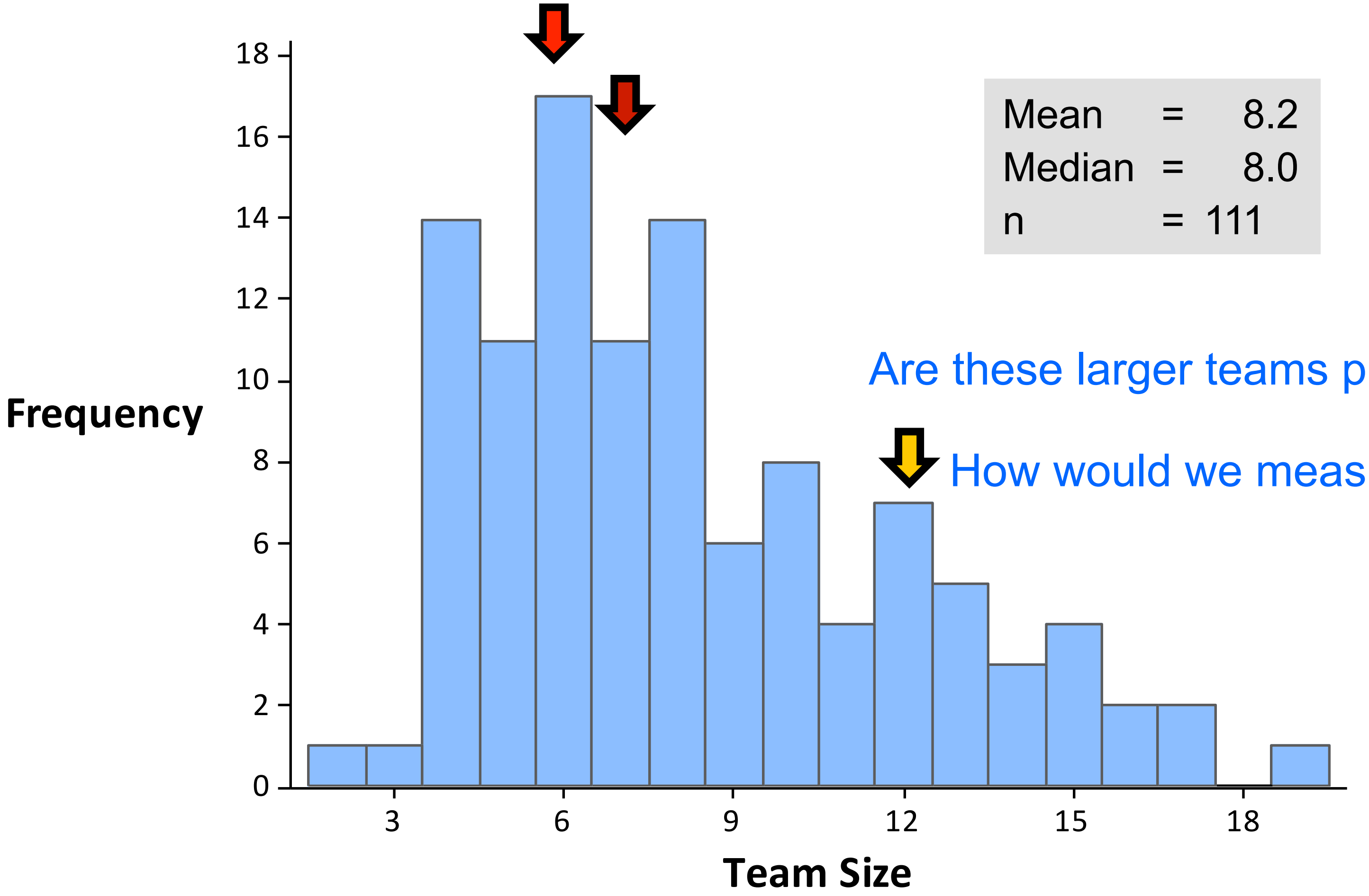
Team Size



Mean = 8.2
Median = 8.0
n = 111

Are these larger teams productive?

Team Size

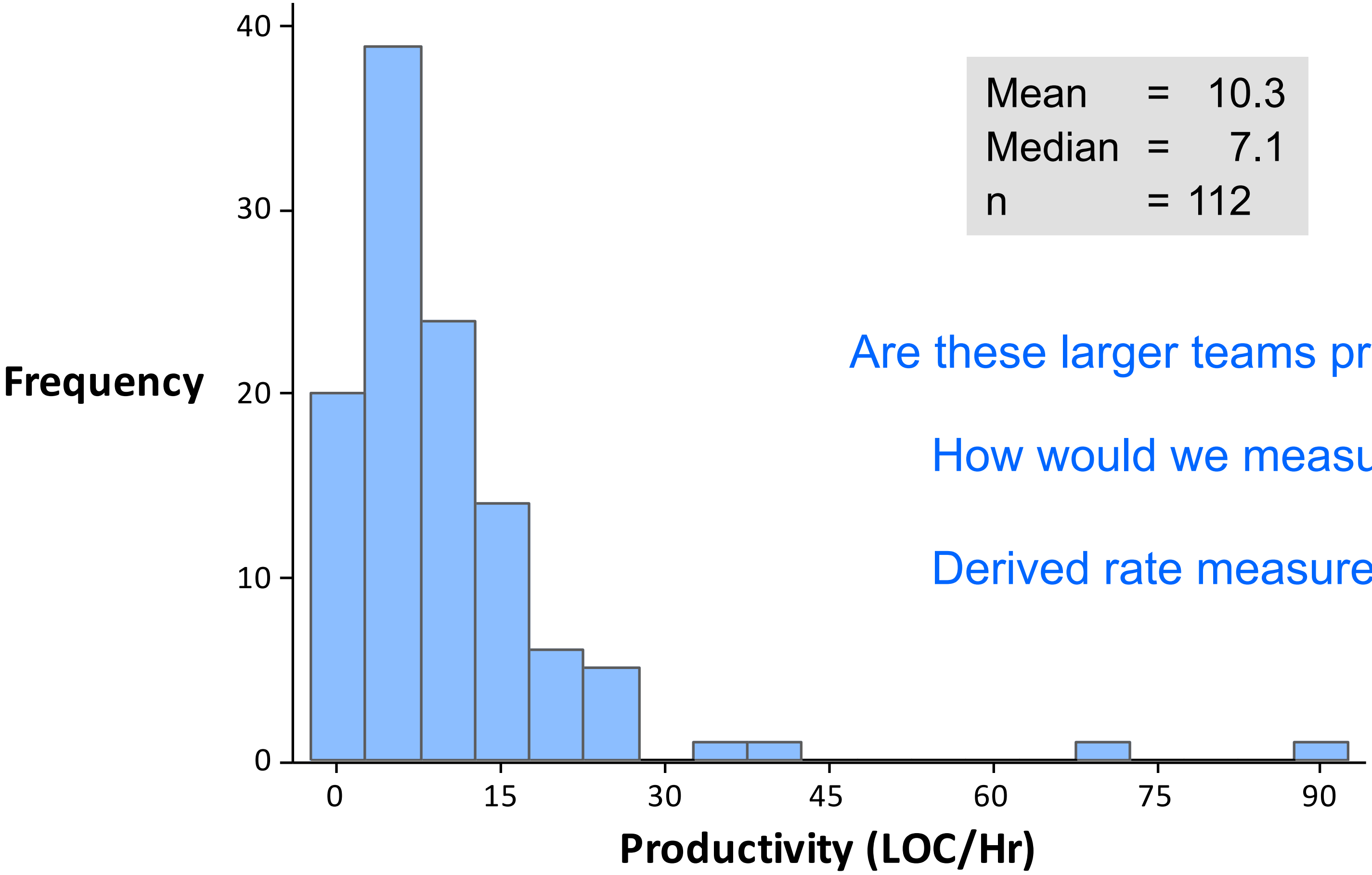


Mean = 8.2
Median = 8.0
n = 111

Are these larger teams productive?

How would we measure?

Code Production Rate



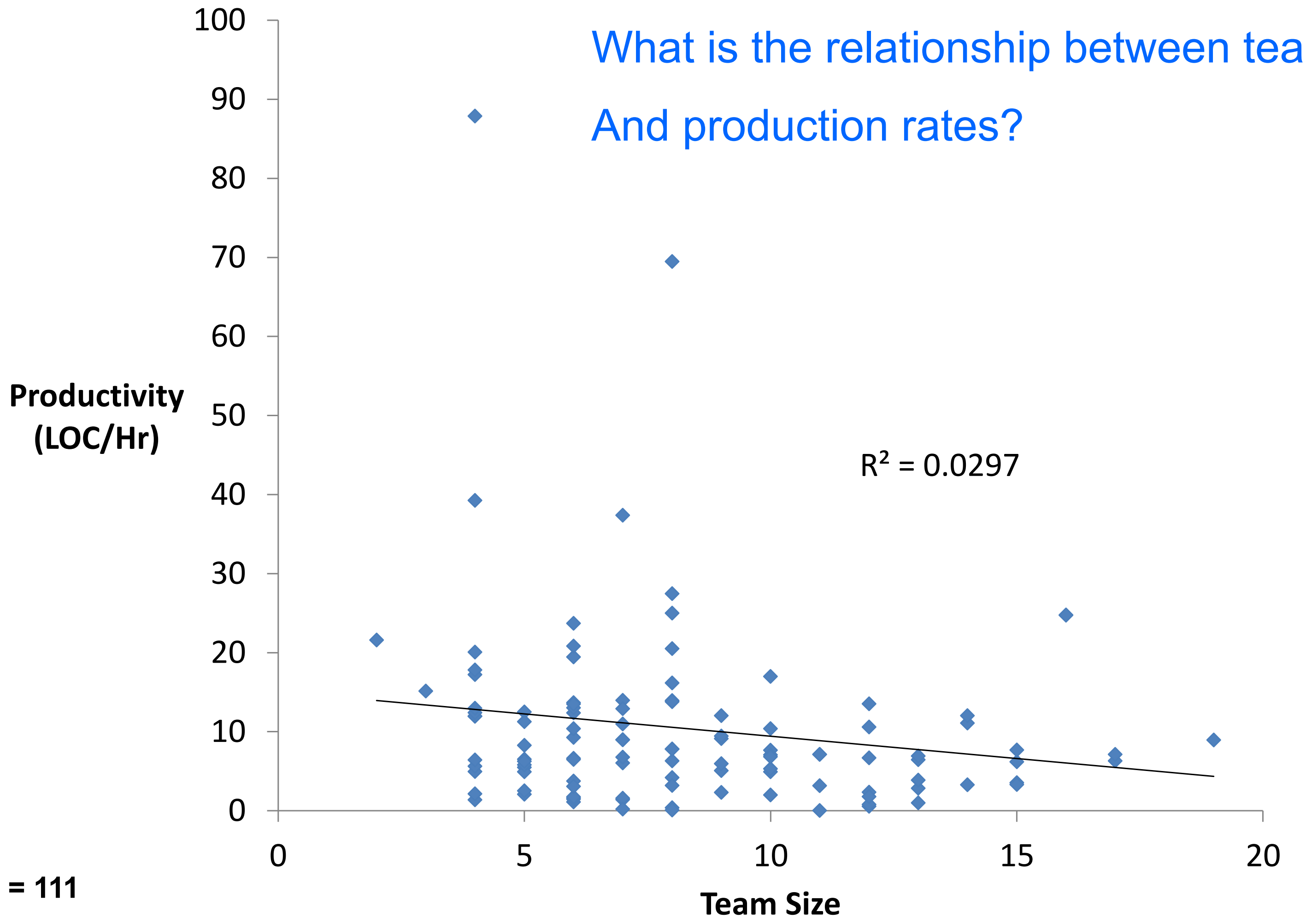
Mean	=	10.3
Median	=	7.1
n	=	112

Are these larger teams productive?

How would we measure?

Derived rate measure

Team Size vs. Productivity

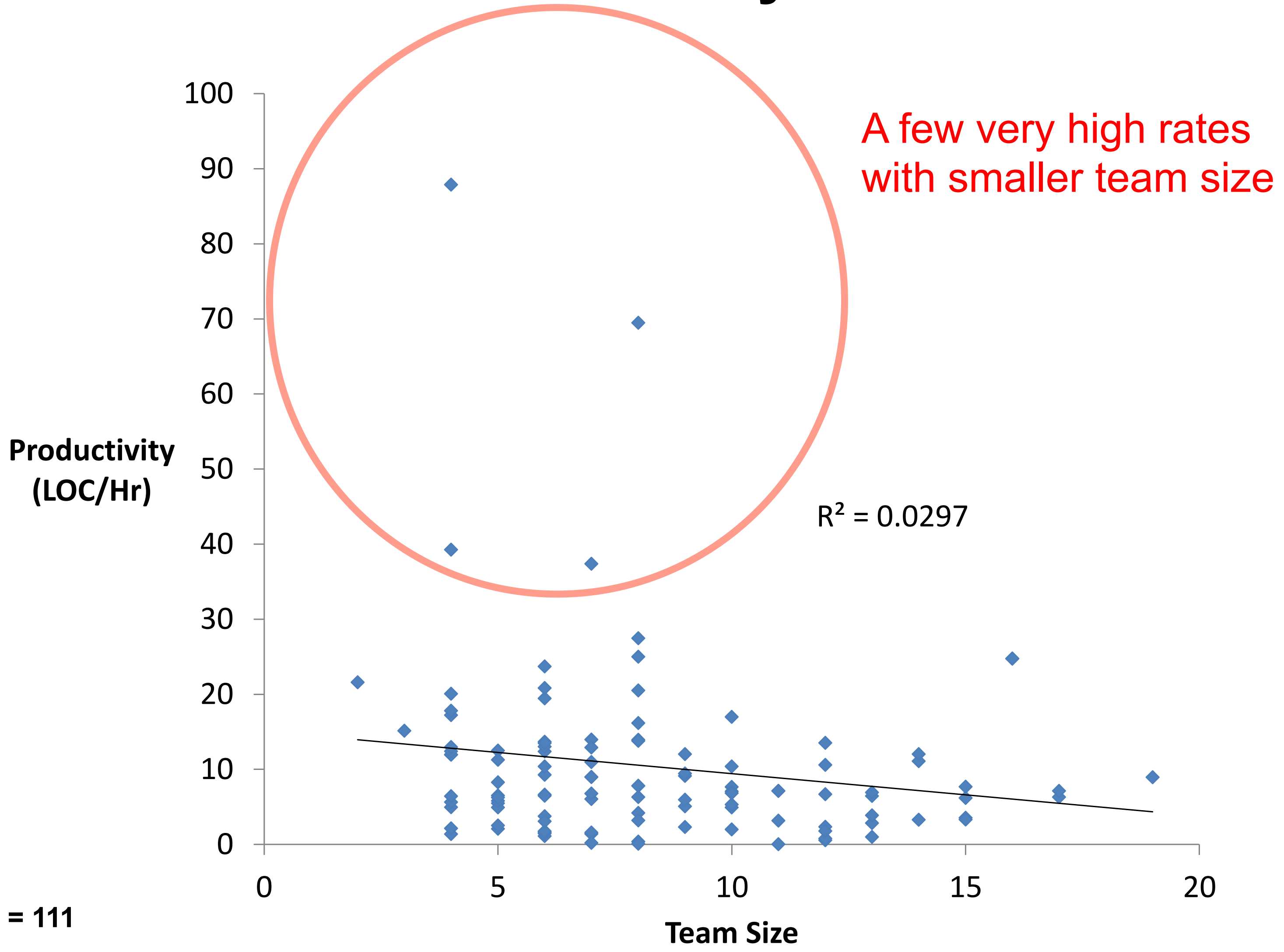


What is the relationship between team size?

And production rates?

n = 111

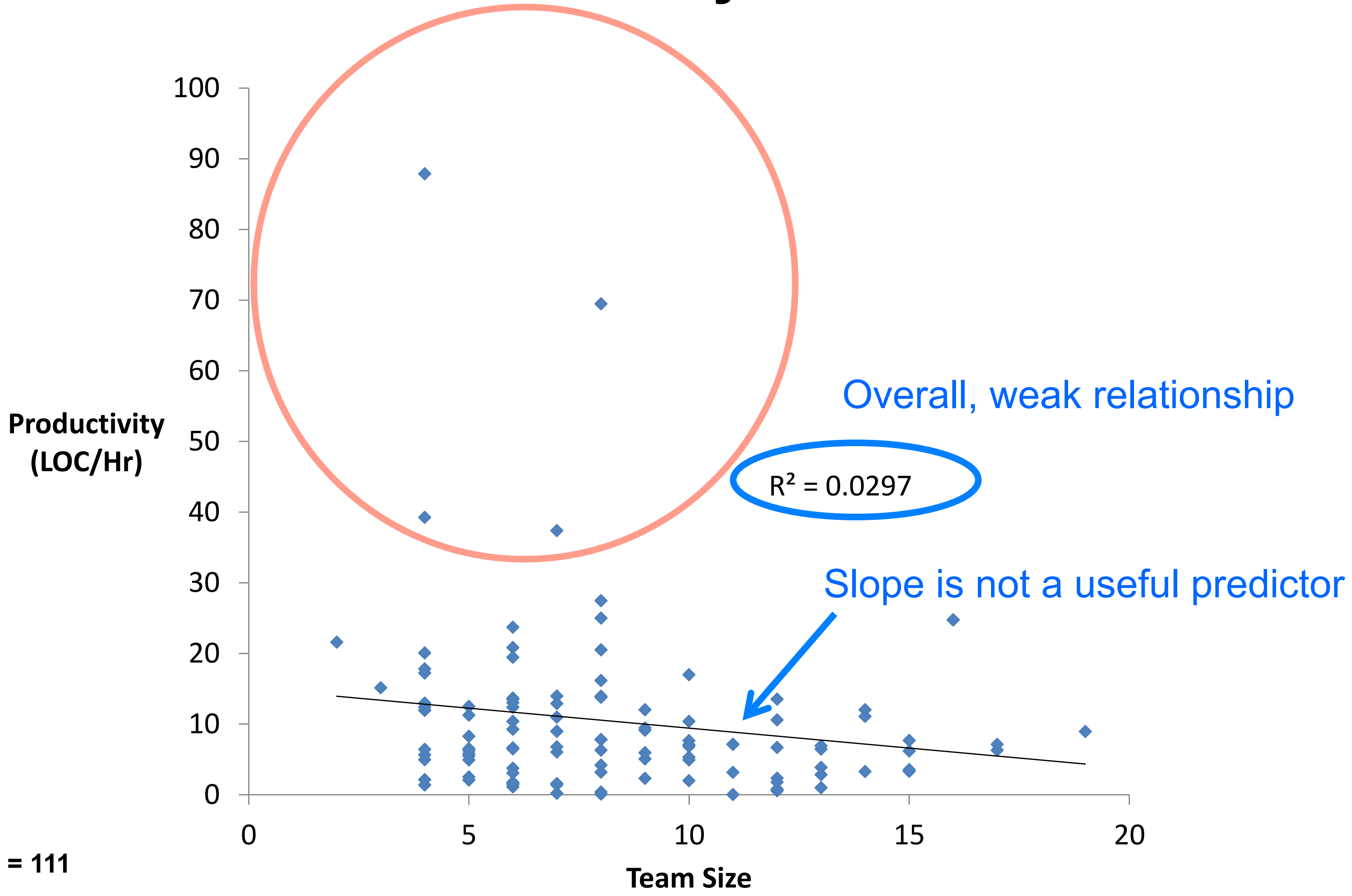
Team Size vs. Productivity



A few very high rates with smaller team size

n = 111

Team Size vs. Productivity

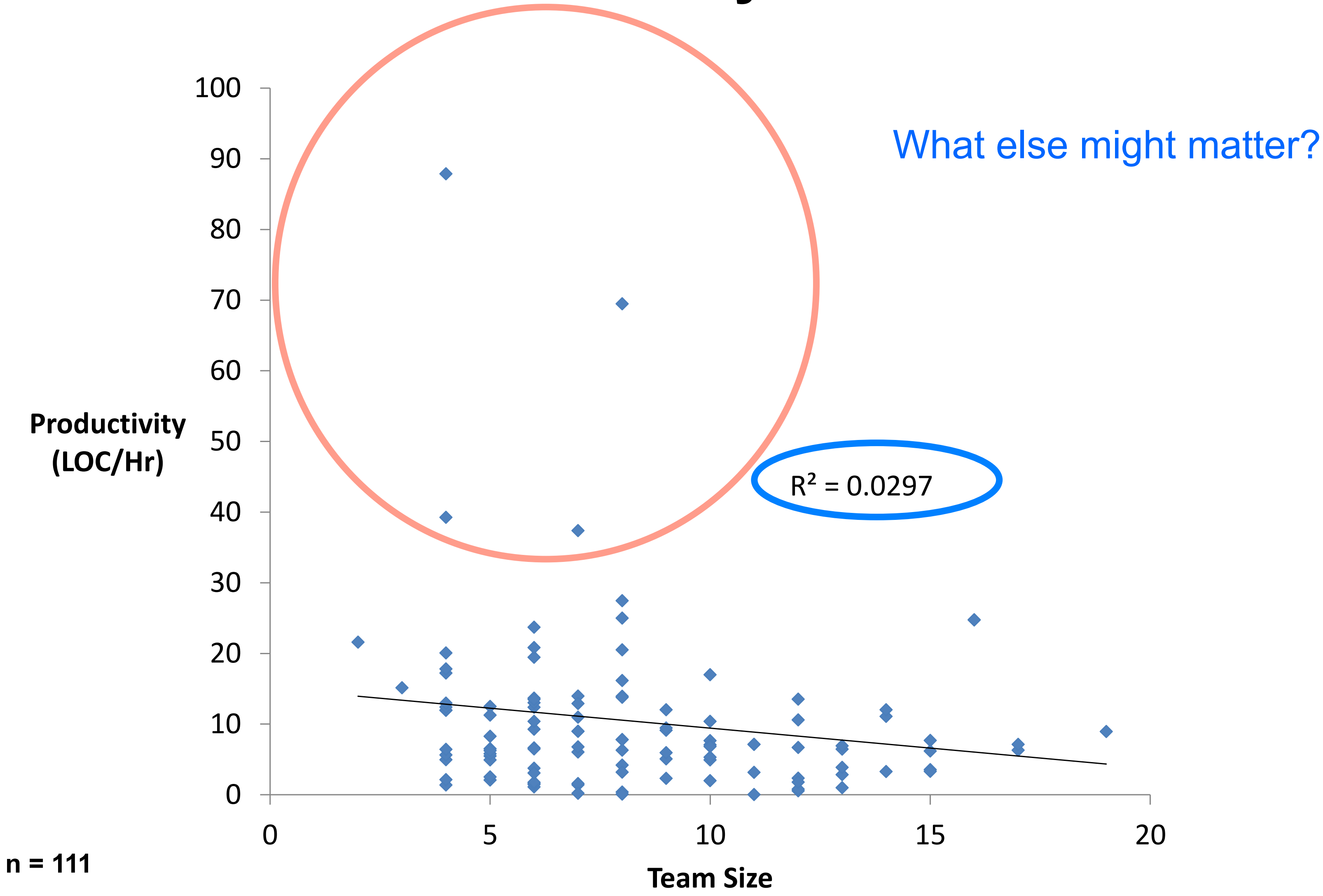


Overall, weak relationship

Slope is not a useful predictor

n = 111

Team Size vs. Productivity



What else might matter?

n = 111

What do you think is the average number of *task hours* that a team member spends during a week?

□ 35 to 40 hours

□ 30 to 35 hours

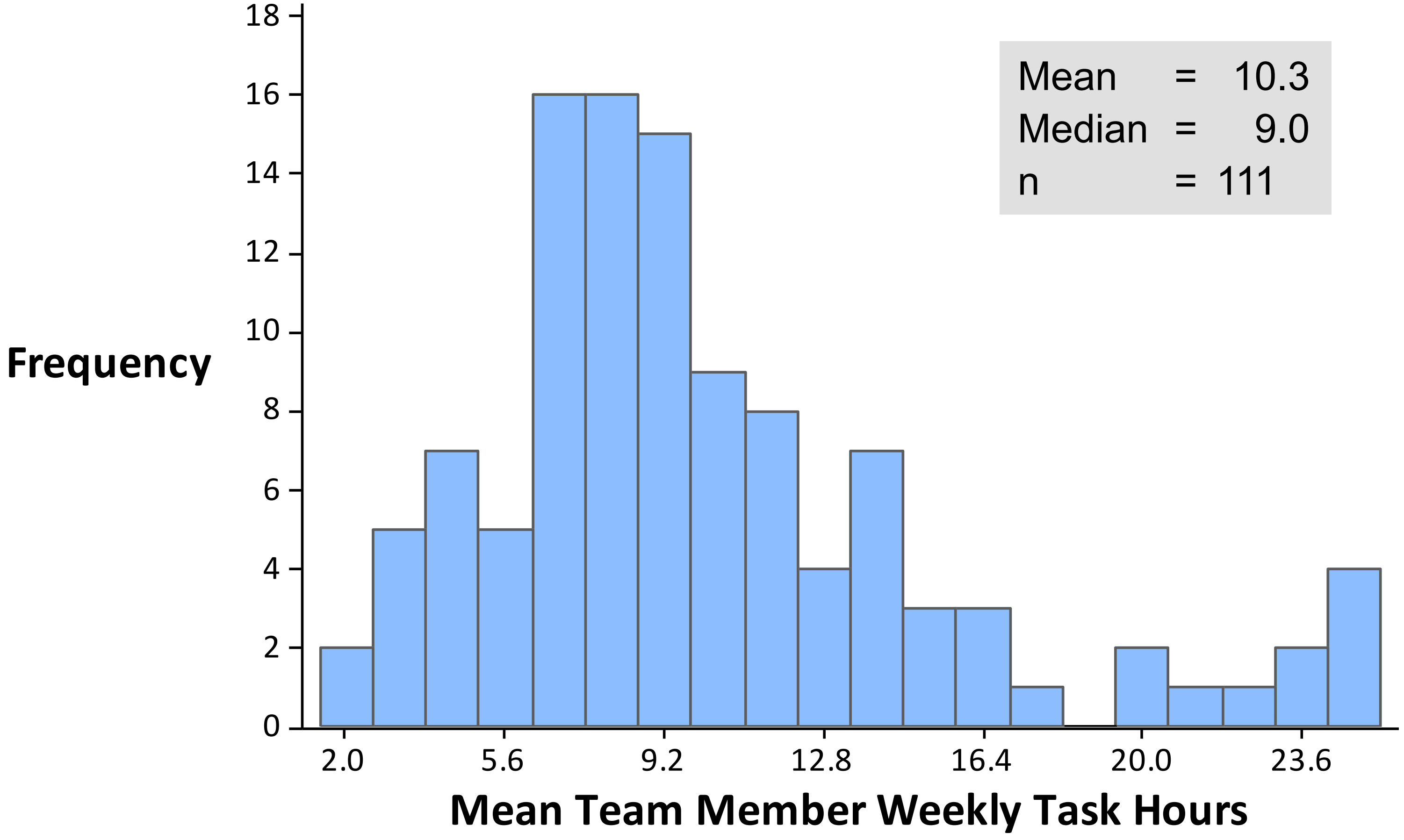
□ 25 to 30 hours

□ 20 to 25 hours

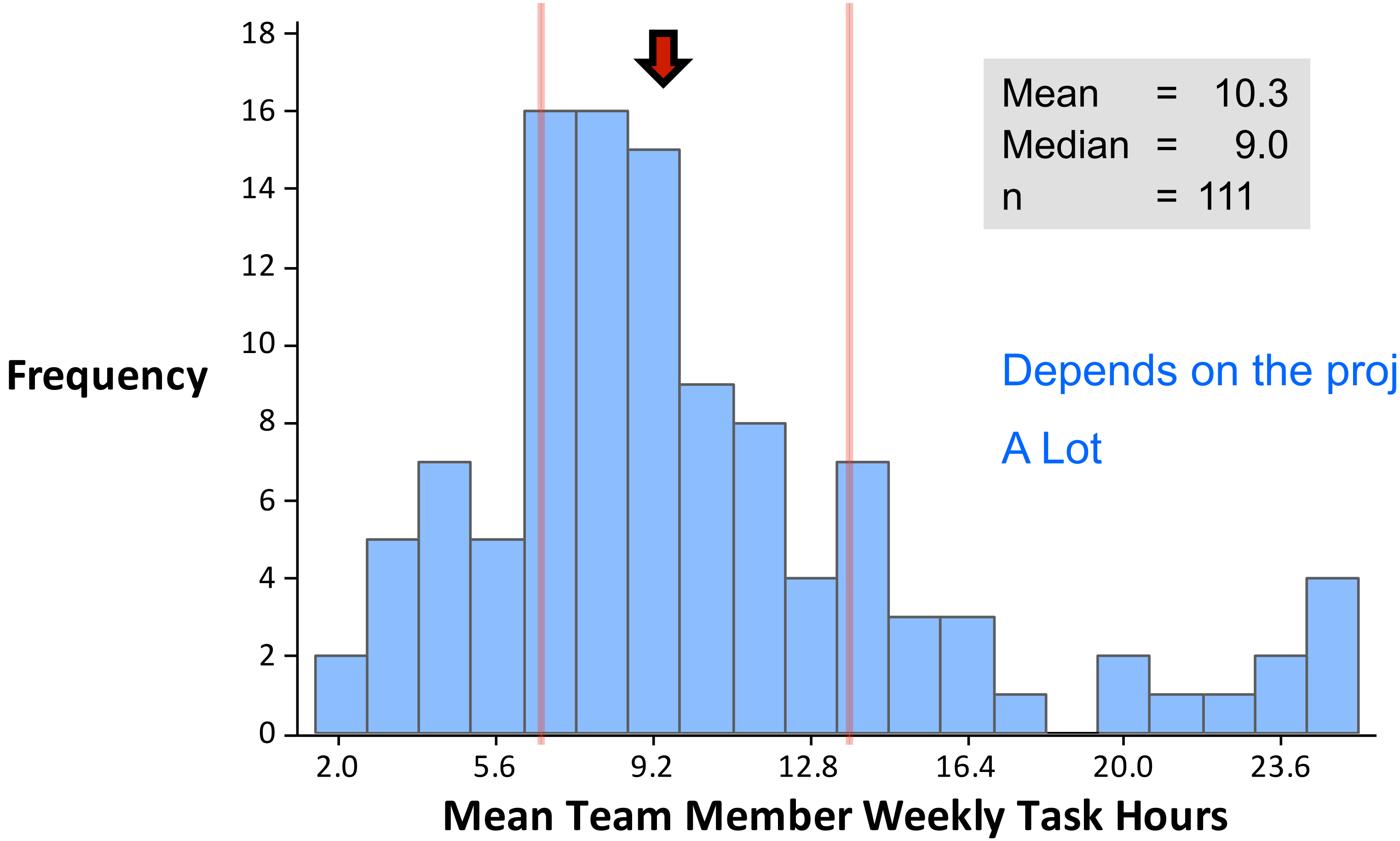
□ 15 to 20 hours

□ less than 15 hours

Mean Team Member Weekly Task Hours



Mean Team Member Weekly Task Hours



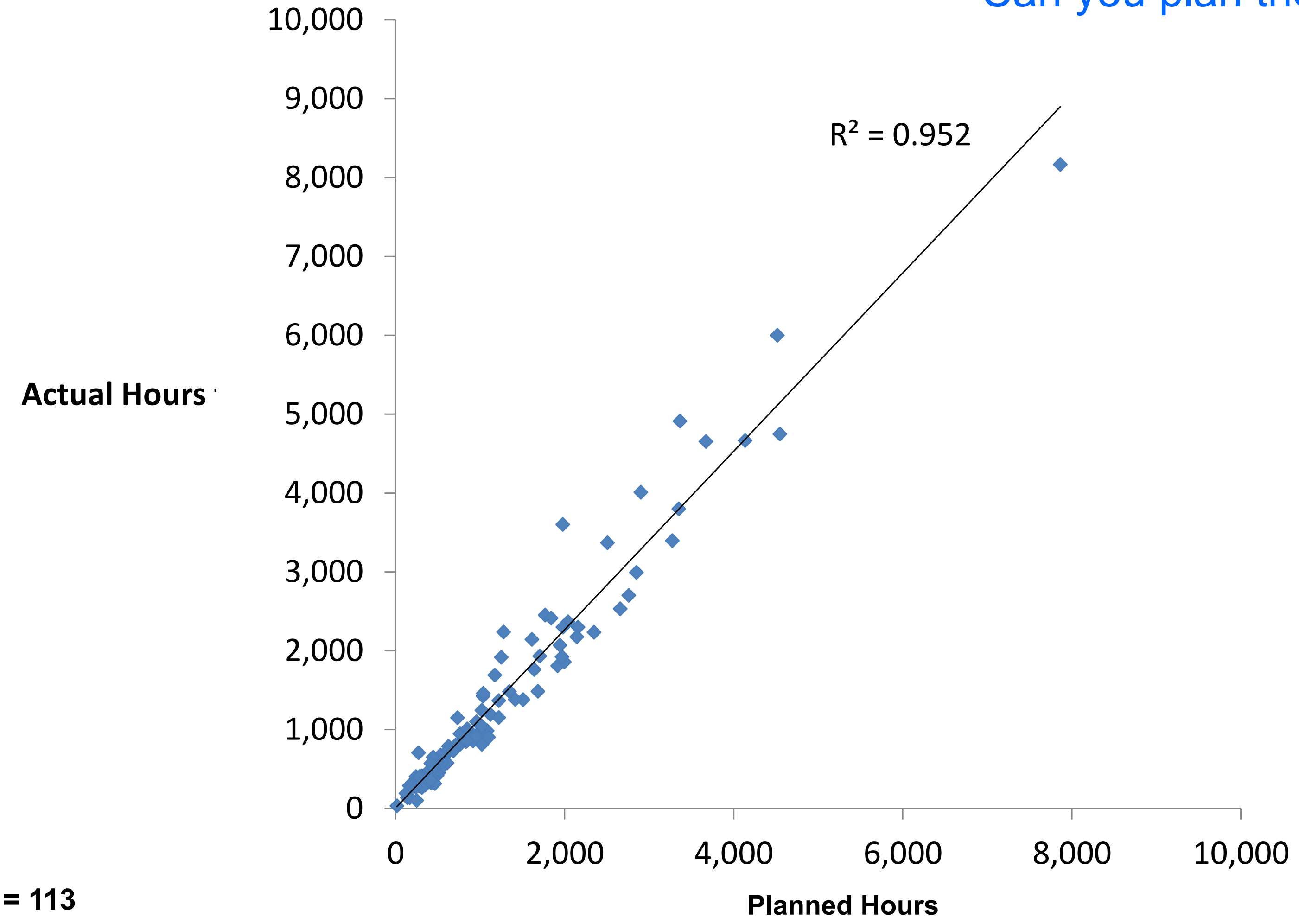
Mean = 10.3
Median = 9.0
n = 111

Depends on the project

A Lot

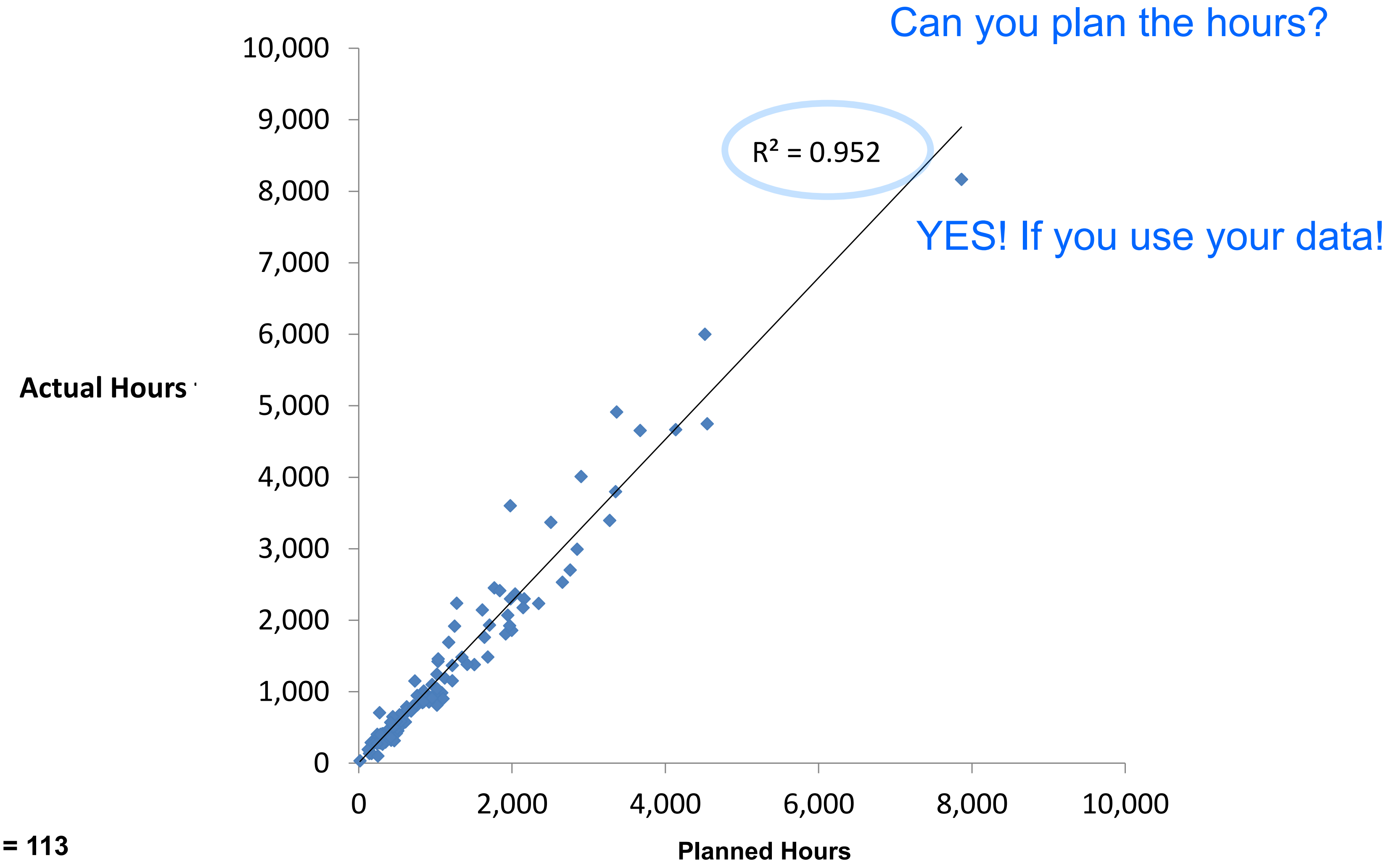
Plan Vs. Actual Hours

Can you plan the hours?



n = 113

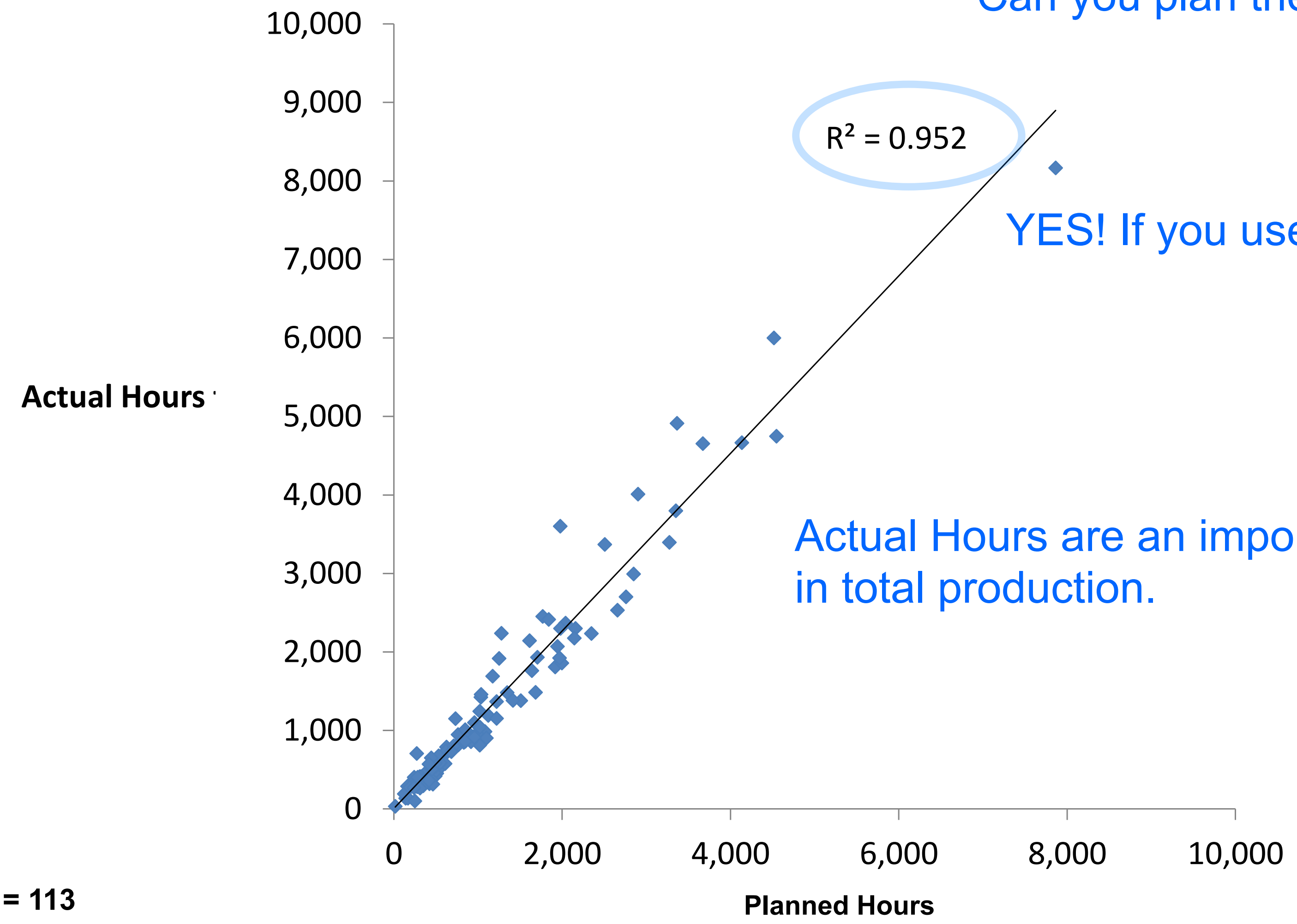
Plan Vs. Actual Hours



n = 113

Plan Vs. Actual Hours

Can you plan the hours?



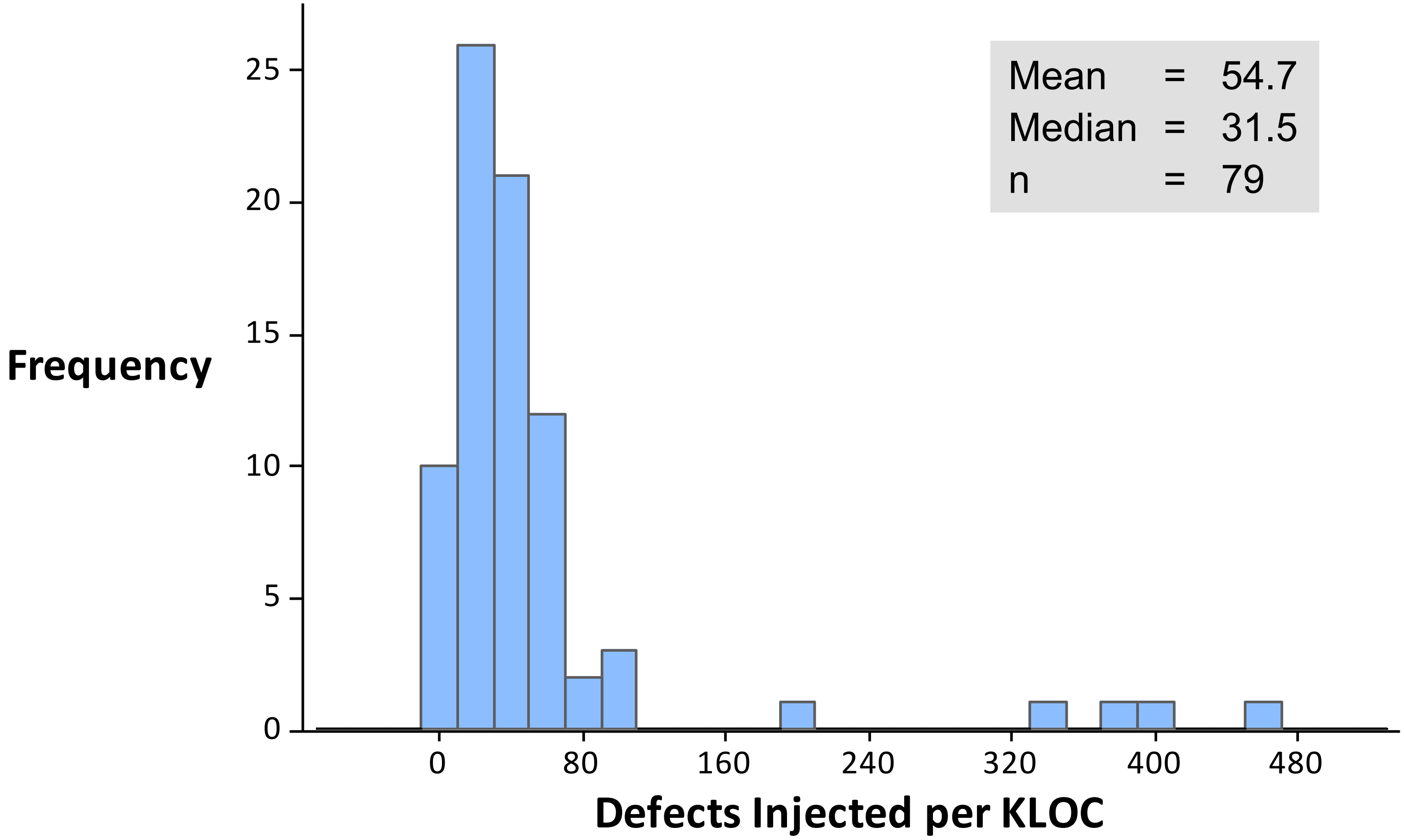
YES! If you use your data!

Actual Hours are an important factor in total production.

n = 113

Let’s Looks at Some Quality-Based Profiles

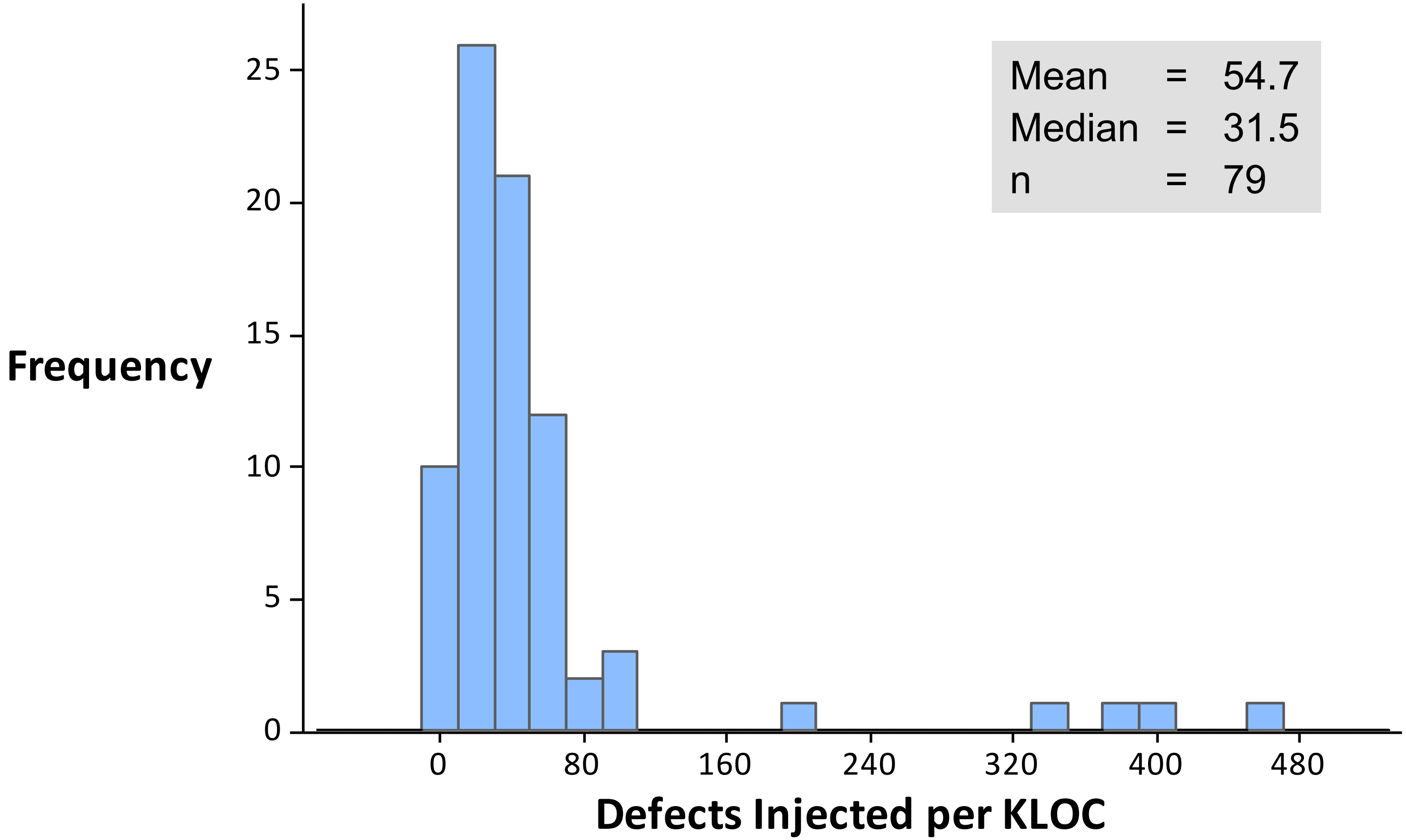
Total Defects Injected Per KLOC



Mean	=	54.7
Median	=	31.5
n	=	79

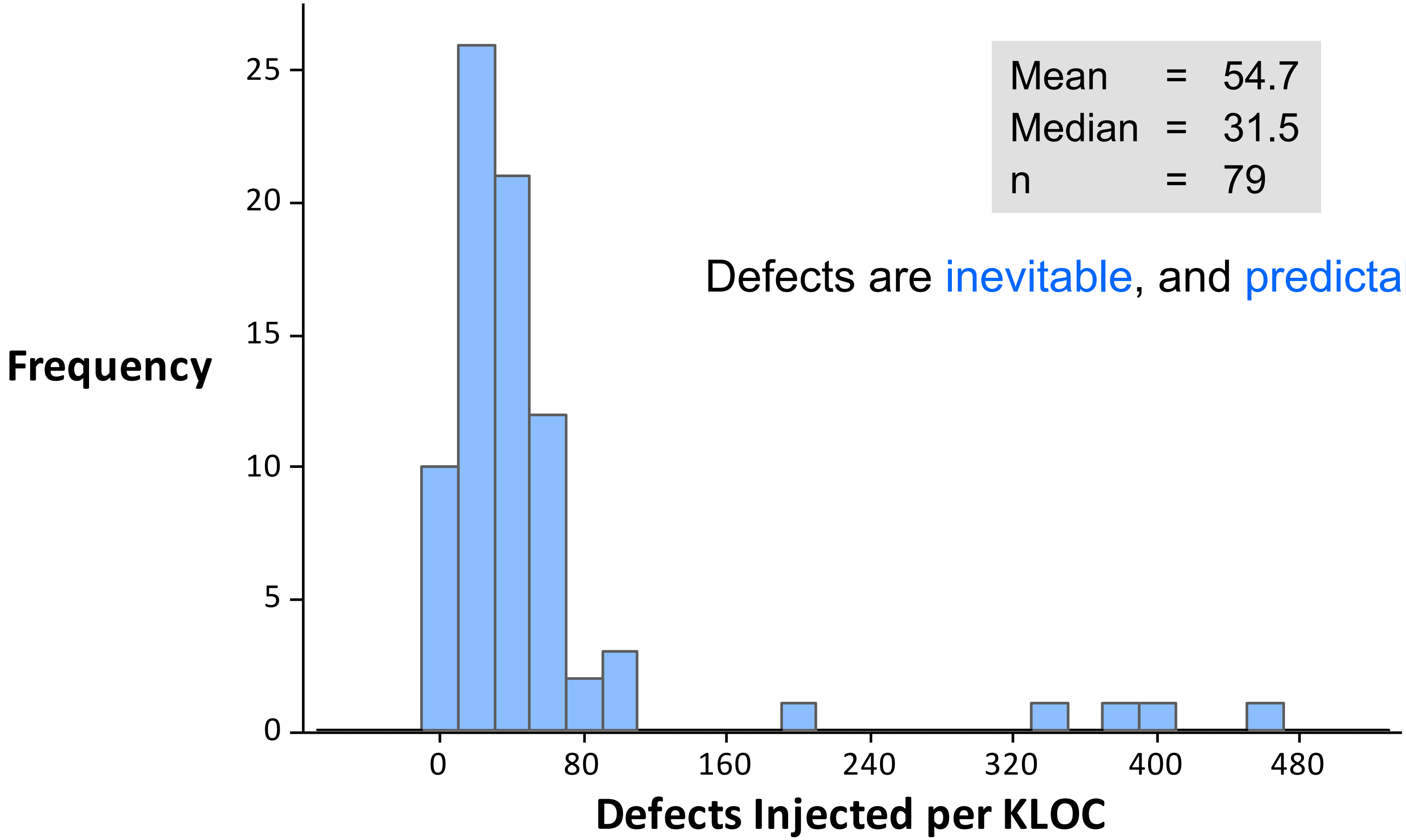
Total Defects Injected Per KLOC

To ~~E~~error is human.



Total Defects Injected Per KLOC

To ~~E~~error is human.

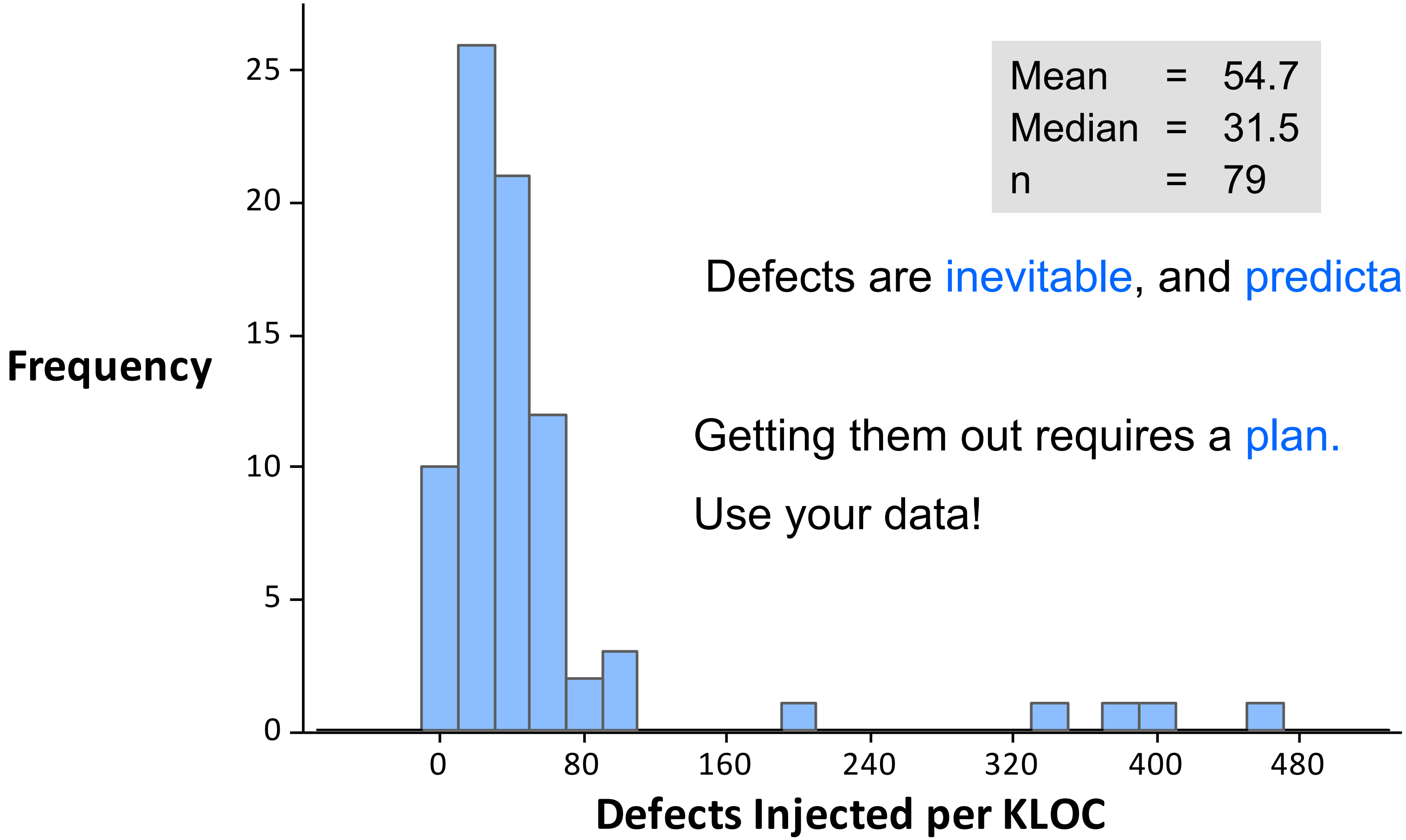


Mean = 54.7
Median = 31.5
n = 79

Defects are inevitable, and predictable.

Total Defects Injected Per KLOC

To ~~E~~error is human.



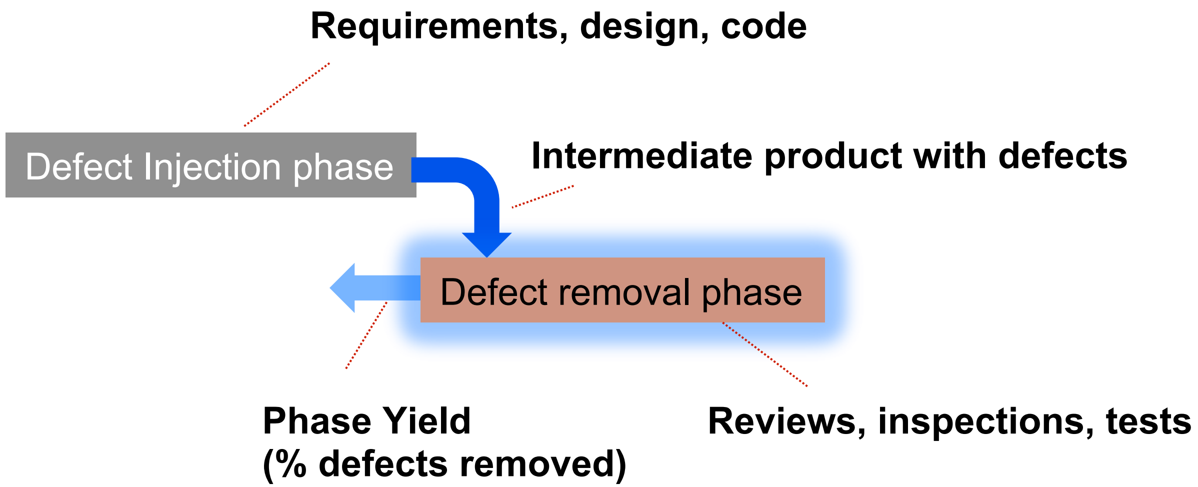
Mean	=	54.7
Median	=	31.5
n	=	79

Defects are inevitable, and predictable.

Getting them out requires a plan.

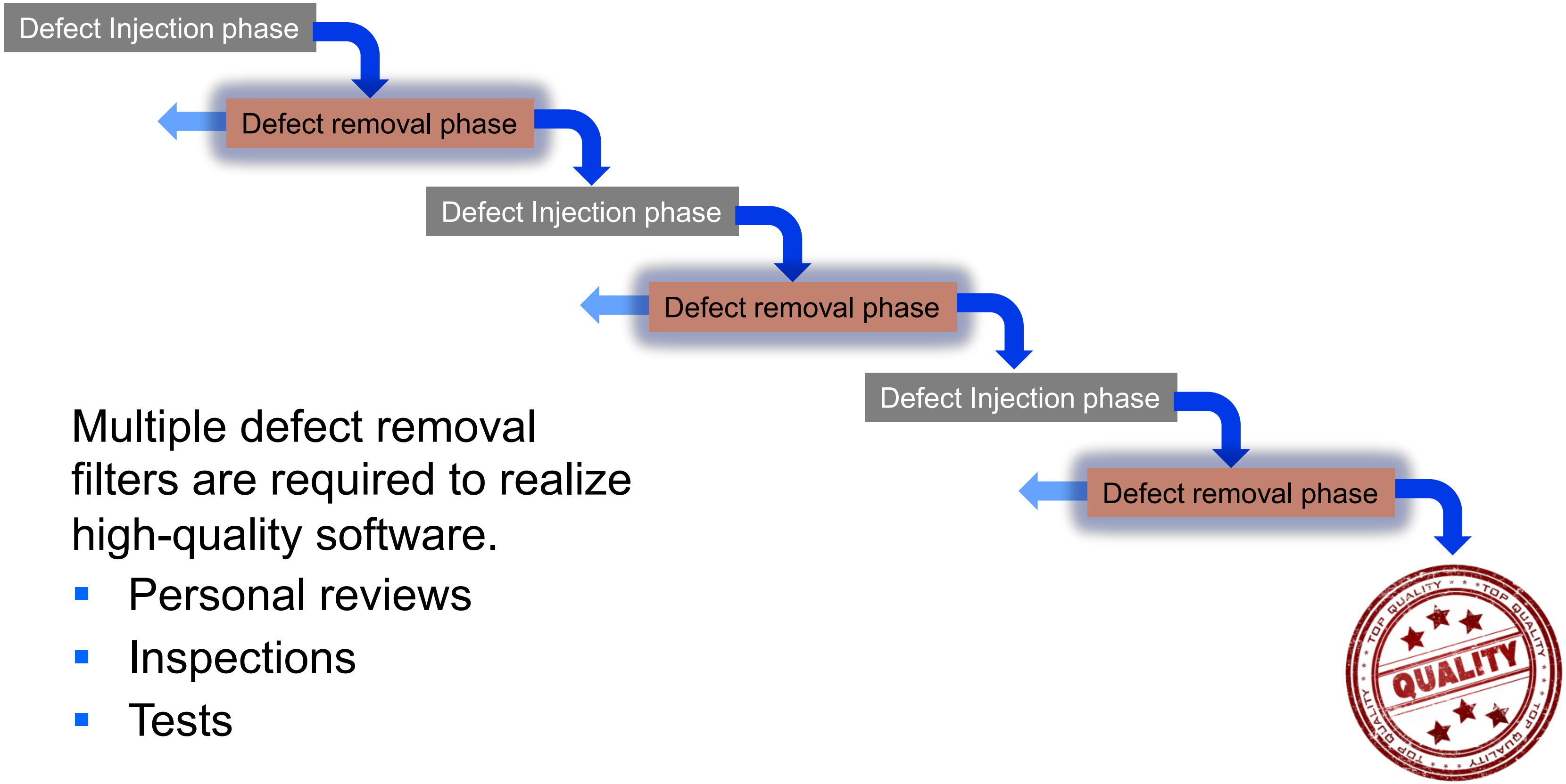
Use your data!

Injection and Removal of Defects



Requirements, design, code
Defect Injection phase
Intermediate product with defects
Defect removal phase
Phase Yield
(% defects removed)
Reviews, inspections, tests

Multiple Defect Removal Filters Required



Defect Injection phase

Defect removal phase

Defect Injection phase

Defect removal phase

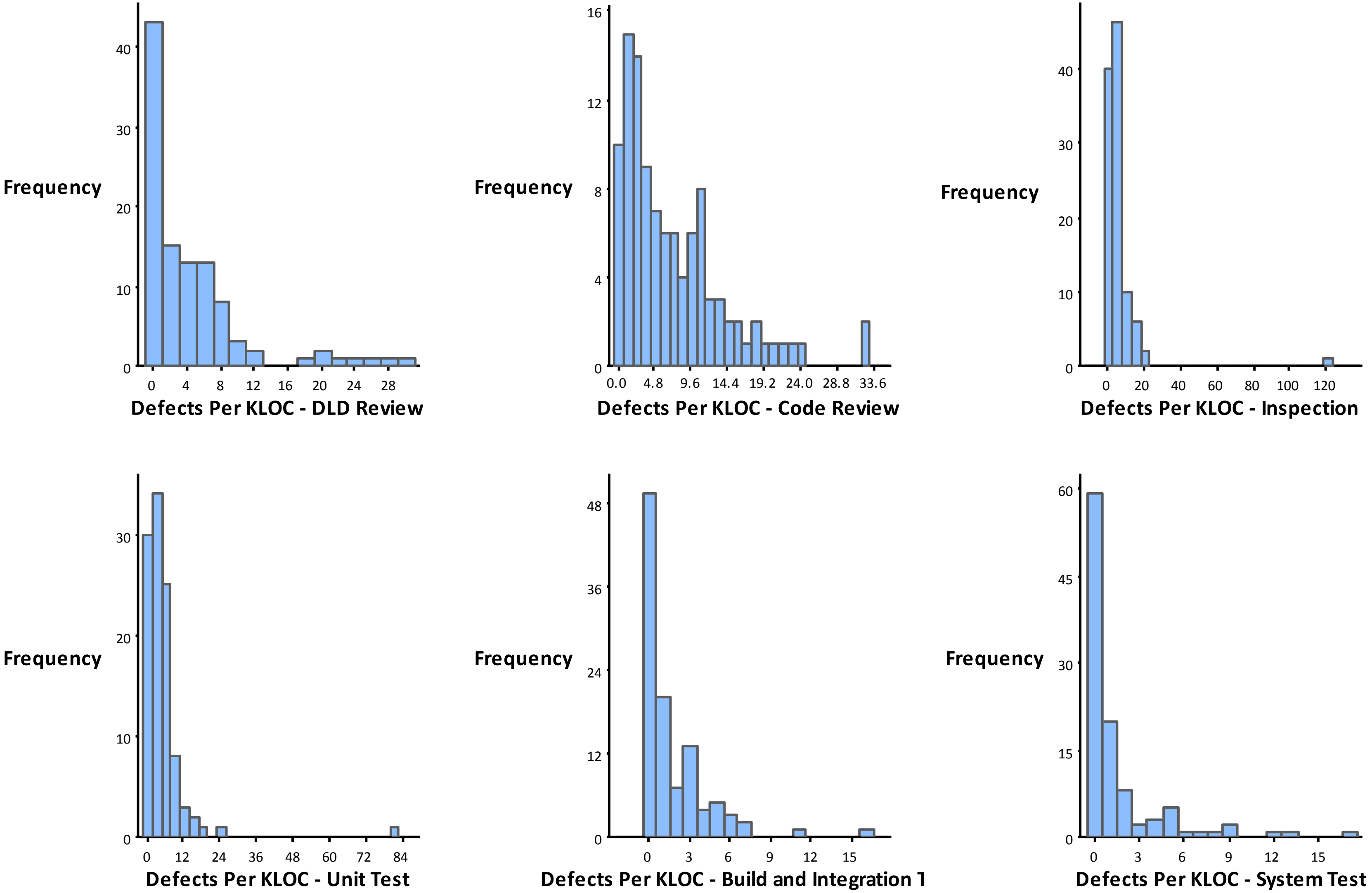
Defect Injection phase

Defect removal phase

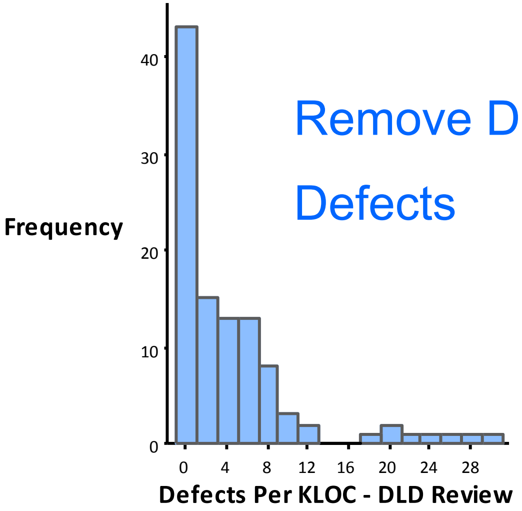
Multiple defect removal filters are required to realize high-quality software.

- Personal reviews
- Inspections
- Tests

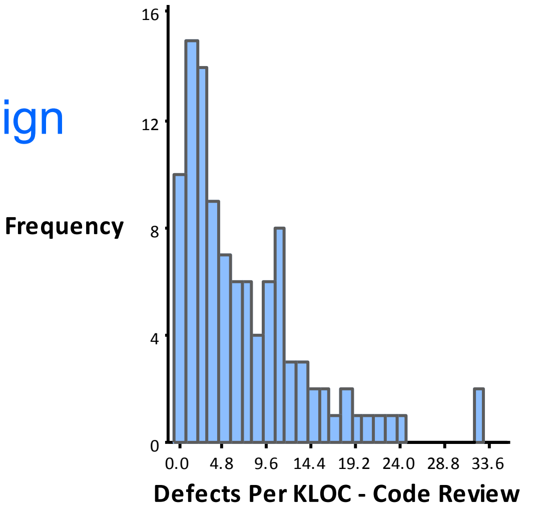
Defect Density - Summary

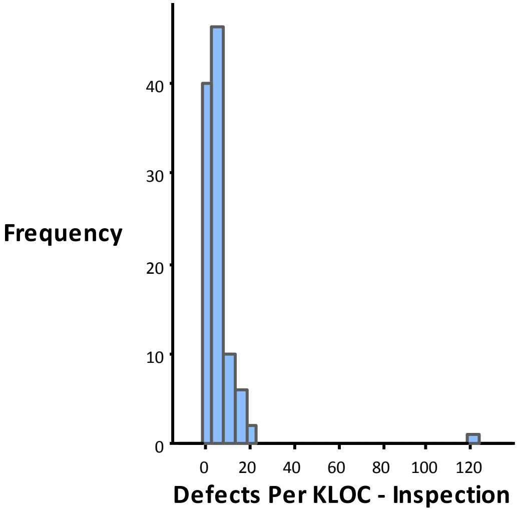


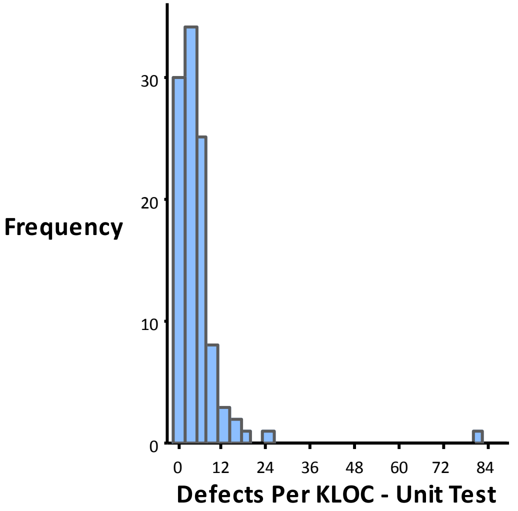
Defect Density - Summary

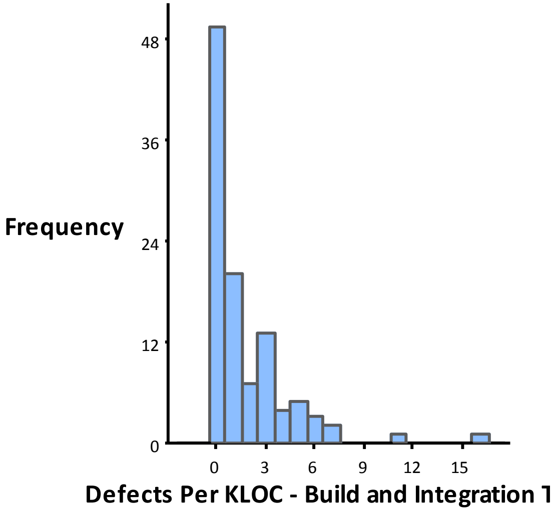


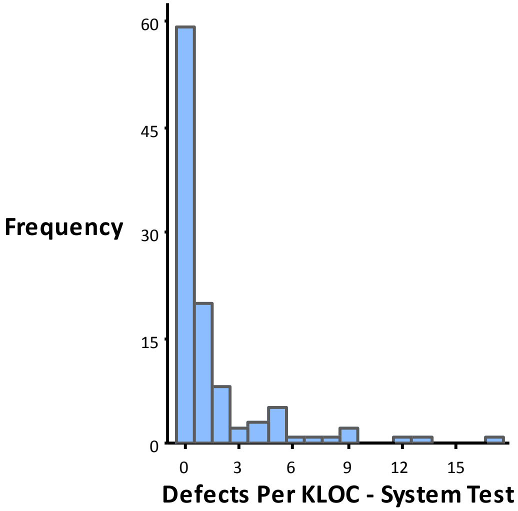
Remove Design Defects



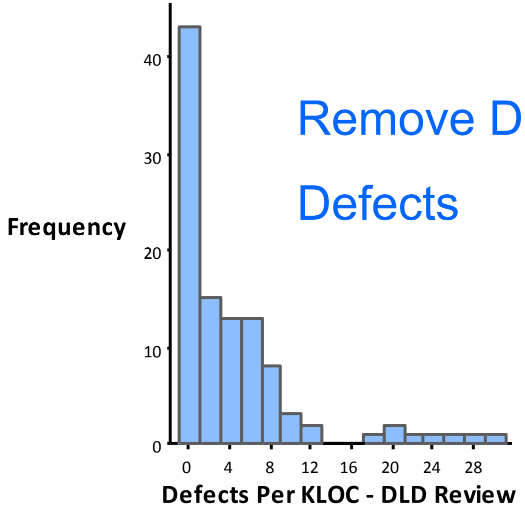




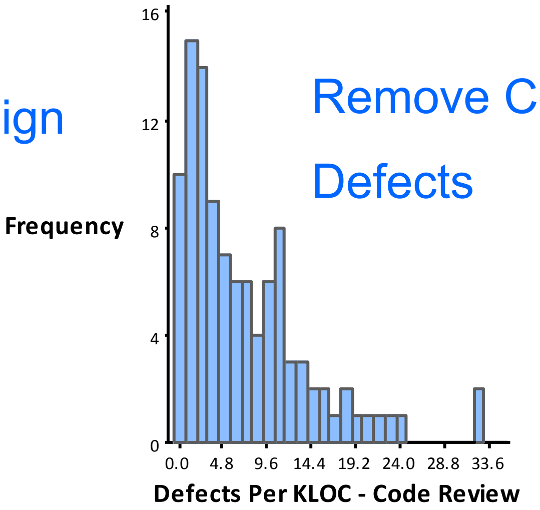




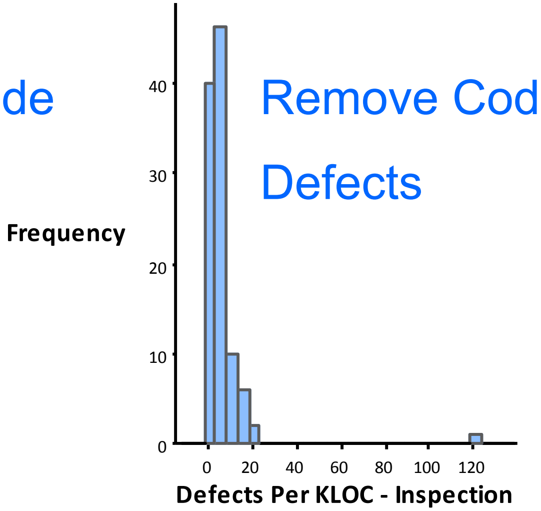
Defect Density - Summary



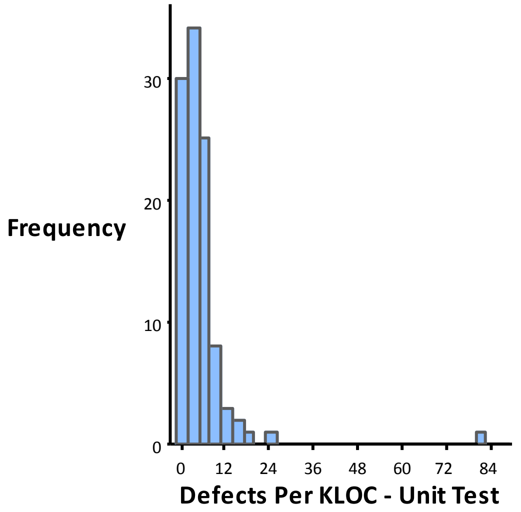
Remove Design Defects

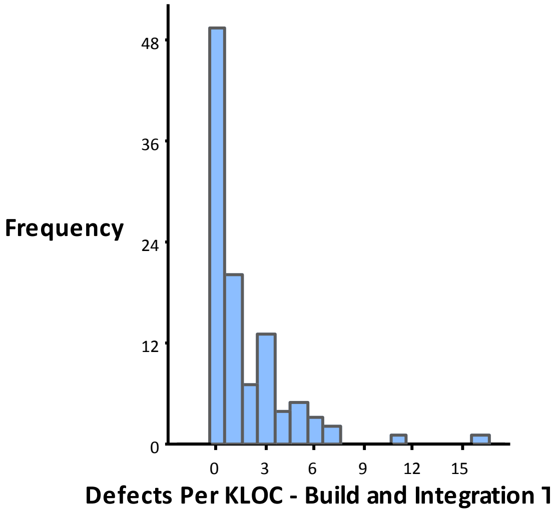


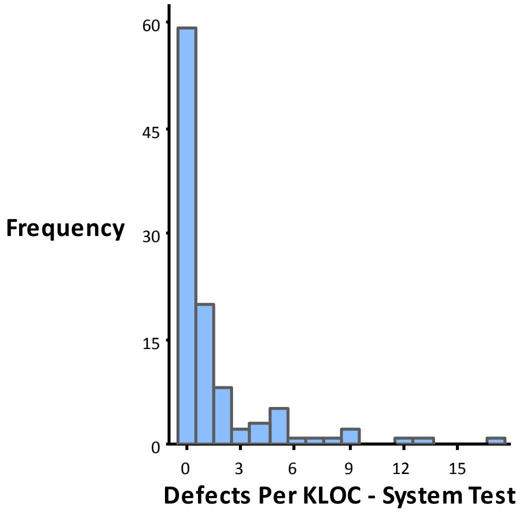
Remove Code Defects



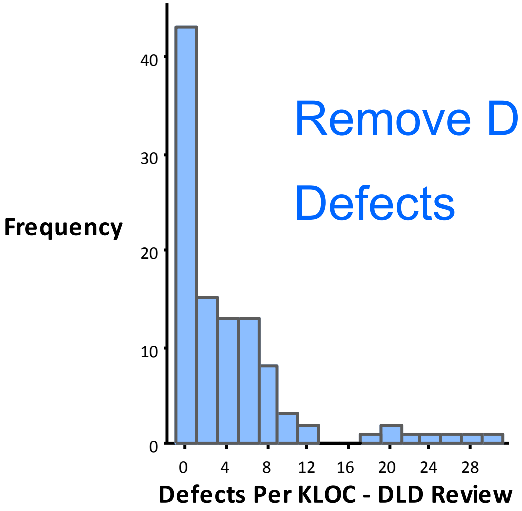
Remove Code Defects



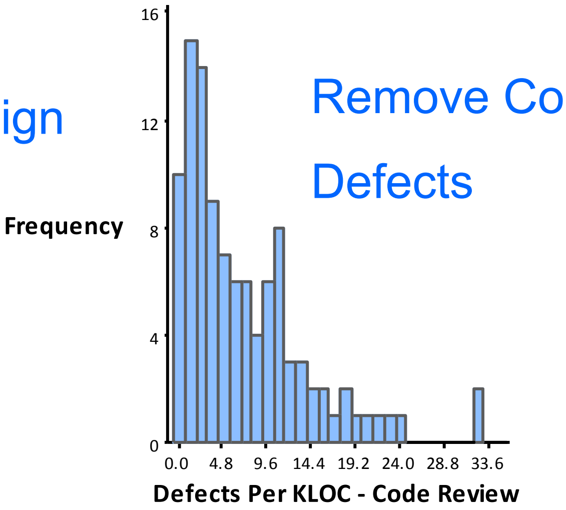




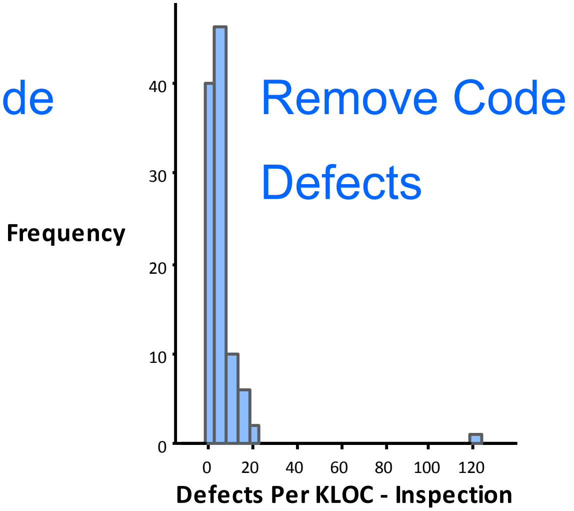
Defect Density - Summary



Remove Design Defects

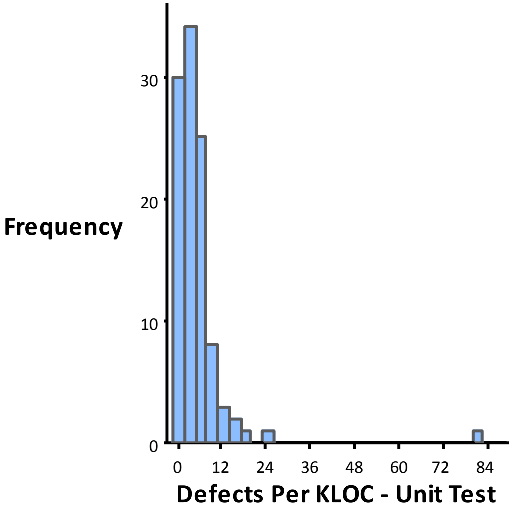


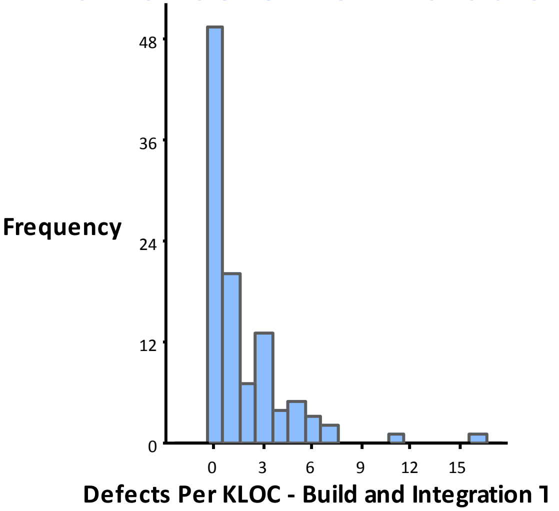
Remove Code Defects

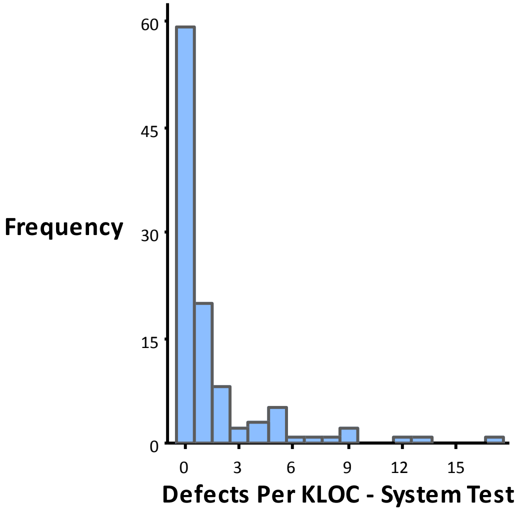


Remove Code Defects

Test Removes Other Defects

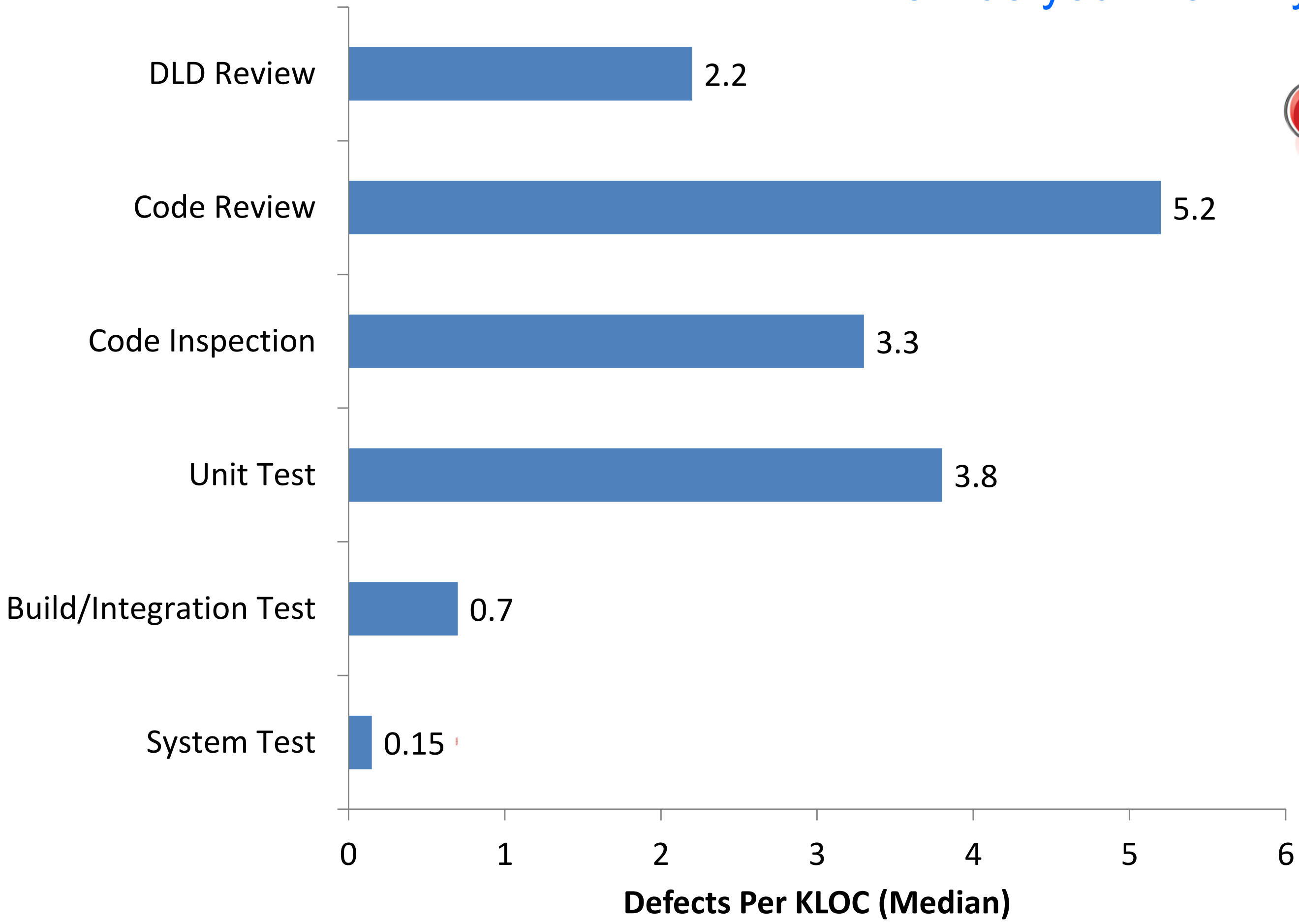






Defect Removal Density –
Median of Defects Per KLOC

How do you know if you are on track?



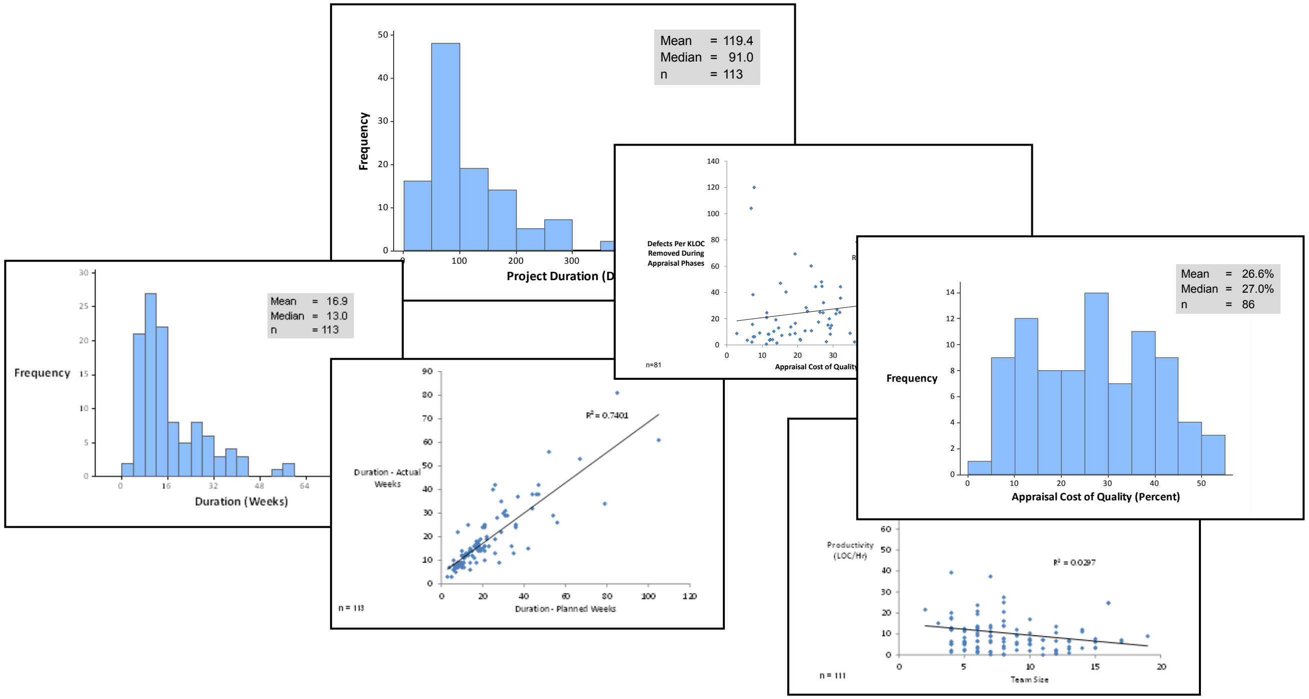
< 5

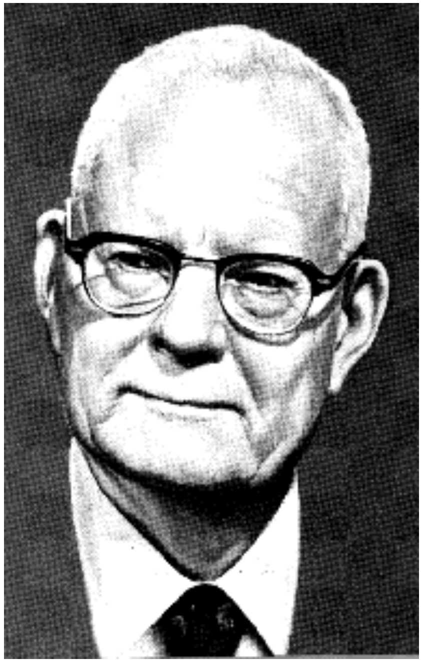
< 0.5

0.2

All project performance charts are available as a download with today’s webinar.

These include charts not presented in this webinar.





W. Edwards Deming

In God we trust ...

All others bring us Data!

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0001164

Contact Information

Mark Kasunic
Senior Member of Technical Staff
Software Engineering & Acquisition Practices
Software Solutions Division
Telephone: +1 412-268-5863
Email: info@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

Customer Relations
Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257

TSP SYMPOSIUM 2014

Going Beyond Methodology to Maximize Performance

9th Annual Software Engineering Institute (SEI) Team Software Process (TSP) Symposium

www.sei.cmu.edu/tspsymposium/2014/

Call for Submissions Now Open

Abstracts for Presentations and Papers Are Being Accepted Through May 16, 2014

Software Engineering Institute
Carnegie Mellon University